

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Návrh a realizace GSM datového komunikátoru pro IQRF sítě s ohledem na spotřebu elektrické energie

Design and Implementation of GSM Data Communicator for IQRF Networks with Respect to Energy Consumption

Zadání diplomové práce

Student:

Bc. Jonáš Stýskala

Studijní program:

N2649 Elektrotechnika

Studijní obor:

2612T041 Řídicí a informační systémy

Téma:

Návrh a realizace GSM datového komunikátoru pro IQRF sítě s ohledem
na spotřebu elektrické energie

Design and Implementation of GSM Data Communicator for IQRF
Networks with Respect to Energy Consumption

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Popis technologie IQRF a mikrokontroléru NXP Kinetis řady L.
2. Popis možností přenosu dat ze sítě IQRF na datové uložení.
3. Návrh a realizace hardware IQRF/GSM komunikátoru s využitím mikrokontroléru NXP Kinetis řady L.
4. Návrh a realizace firmware pro daný komunikátor.
5. Zprovoznění komunikace mezi IQRF/ GSM komunikátorem a datovým uložením.
6. Vytvoření komunikačního protokolu pro servisní nastavení komunikátoru.
7. Testování vytvořeného zařízení s ohledem na spotřebu elektrické energie.
8. Zhodnocení výsledků a závěr.

Seznam doporučené odborné literatury:

- [1] JACKO, Róbert. *Energeticky soběstačný bezdrátový modul pro senzorické aplikace*. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. Dostupné také z: <http://hdl.handle.net/11012/32913>.
- [2] BRTNÍK, Bohumil a David MATOUŠEK. *Algoritmy číslicového zpracování signálů*. Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-400-2.
- [3] DOSTÁLEK, Libor a Alena KABELOVÁ. *Velký průvodce protokoly TCP/IP a systémem DNS*. 2. vyd. Computer Press Praha, 2000. 426 s. ISBN 80-7226-323-4.
- [4] NXP Semiconductors. *KL28Z Reference Manual*. 2016. [online] Dostupné z: <http://www.nxp.com>.
- [5] MICRORISC s.r.o. *SPI Implementation in IQRF For (DC)TR-7xD*. 2016. [online] Dostupné z: <http://www.iqrf.org>.
- [6] MICRORISC s.r.o. *IQRF Quick Start Guide For IQRF OS v3.07D and higher*. 2017. [online] Dostupné z: <http://www.iqrf.org>.
- [7] MICRORISC s.r.o. *IQRF OS Operating System Version 4.00D for (DC)TR-7xD - User's Guide*. 2017. [online] Dostupné z: <http://www.iqrf.org>.
- [8] MICRORISC s.r.o. *IQRF DPA Framework Technical Guide v3.00*. 2017. [online] Dostupné z: <http://www.iqrf.org>.
- [9] MICRORISC s.r.o. *IQRF Cloud Technical guide*. 2015. [online] Dostupné z: <http://www.iqrf.org>.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Martin Pieš, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2019



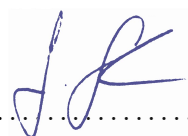
doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2019

A handwritten signature in blue ink, consisting of stylized initials and a surname, positioned above a dotted line.

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli. Jedná se především o vedoucího této práce pana Ing. M. Pieše, Ph.D., který svými zkušenostmi a podporou pomohl k vyřešení ne málo problémů vztahujících se k realizaci komunikátoru. Navíc poskytoval během celé realizace rady, díky nimž jsem mohl nabýt nových znalostí a zkušeností v oblasti návrhu a výroby elektronických zařízení. Tato podpora byla nedocenitelná a nesmírně si jí vážím. Dále také panu Ing. M. Stankušovi, Ph.D. za vstřícnost a poskytnutou podporu v rámci firmware.

Abstrakt

Tato diplomová práce je zaměřena na realizaci elektronického zařízení, které bude představovat most mezi IQRF MESH sítí a mezi dostupným datovým úložištěm. Zařízení bude svou činností sbírat data z bezdrátové sítě měřicích zařízení (IQRF) a tato data za pomoci GSM sítě distribuovat směrem k dostupnému datovému úložišti, které bude sloužit také jako poskytovatel dat k prezentaci těchto měření. Výsledné komunikační zařízení bude pracovat s ohledem na spotřebu elektrické energie a předpokládá se, že bude moci být situováno v oblastech bez stálých zdrojů napájení. Splněním těchto požadavků by komunikátor mohl vyplnit mezeru mezi dostupnými zařízeními podobného typu, a to především díky napájecím možnostem - schopnosti pracovat bez stálého zdroje elektrické energie.

Klíčová slova: diplomová práce, IQRF, GSM, MCU, low-power

Abstract

The thesis aims to produce electronic device, which will be a bridge between IQRF MESH network and some available data storage. The device will collect data from the wireless measuring device network (IQRF) and distribute this data to an available data repository using the GSM network. Data repository will also serve as a base storage for presentation of the measurements. The resulting communication device will operate with regard to electricity consumption and is expected to be located in areas without permanent power supplies. By meeting these requirements, the communicator could fill the gap between available devices of a similar type, mainly due to its power options - the ability to operate without a constant power source.

Key Words: master thesis, IQRF, GSM, MCU, low-power

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	10
Seznam tabulek	11
Seznam výpisů zdrojového kódu	12
1 Úvod	13
2 Popis použité technologie	14
2.1 IQRF	14
2.2 NXP mikrokontrolér	18
2.3 GSM modul	20
2.4 Zdroj energie	21
3 Možnosti přenosu dat ze sítě IQRF na datové úložiště	23
3.1 Veřejně dostupná úložiště	23
3.2 Použité úložiště	24
4 Návrh a realizace hardware komunikátoru	25
4.1 Napájení	25
4.2 Zapojení MCU	28
4.3 Zapojení IQRF modulu	30
4.4 Zapojení GSM modulu	30
4.5 Výsledné zařízení	32
5 Návrh a realizace firmware	35
5.1 Struktura firmware	36
5.2 Využité periferie	39
5.3 Základní chování komunikátoru	42
5.4 Nastavení pracovního režimu	44
5.5 Průběh měřicího cyklu	45
5.6 Průběh servisního zásahu	46
5.7 Průběh probuzení komunikátoru	48
5.8 Průběh odesílání dat datovému úložišti	51

6	Komunikace s datovým úložištěm	53
6.1	Protokol servisních a datových zpráv	54
6.2	Zpracování zpráv PHP skriptem	56
7	Servisní konfigurace	58
7.1	Servisní nastavení komunikátoru	58
8	Spotřeba elektrické energie	62
9	Obsluha zařízení	63
9.1	Příprava zařízení pro start	63
9.2	Změna pracovního režimu	63
9.3	Uvedení zařízení do provozu	63
9.4	Reset zařízení	63
9.5	Vypnutí zařízení	64
10	Zhodnocení výsledků a závěr	65
	Literatura	67

Seznam použitých zkratek a symbolů

DPS	– Deska plošných spojů
FW	– Firmware
GPRS	– General Packet Radio Service
GSM	– Groupe Spécial Mobile (Globální Systém pro Mobilní komunikaci)
HW	– Hardware
HWP	– Hardware Profile (Hardware profil)
I ² C	– Inter-Integrated Circuit (Sériová komunikační sběrnice)
IDE	– Integrated Development Enviroment (Integrované vývojové prostředí)
IoT	– Internet of Things (Internet věcí)
IPC	– Inter-Process Communication (Inter-procesní komunikace)
MCU	– Microcontroller Unit (Mikrokontrolér)
OS	– Operating System (Operační systém)
RF	– Radio Frequency (Rádiová frekvence)
RTC	– Real Time Clock (Hodiny reálného času)
RTOS	– Real Time Operating System (Operační systém reálného času)
SDK	– Software Development Kit
SPI	– Serial Peripheral Interface (Sériové periferní rozhraní)
SW	– Software
TCP	– Transmission Control Protocol (Přenosový protokol)
UART	– Universal asynchronous receiver-transmitter (Univerzální asynchronní přijímač-vysílač)
USB	– Universal Serial Bus (Univerzální sériová sběrnice)

Seznam obrázků

1	IQRF modul. [13]	14
2	Struktura programování pod OS. [14]	16
3	Struktura programování využívající DPA. [14]	17
4	Ukázka topologie mesh sítě. [5]	17
5	Struktura přenosu dat mezi IQMESH, IQRF cloud a uživatelskou aplikací. [6]	18
6	Blokový diagram mikrokontrolérů série Kinetis L KL2x. [16]	20
7	Zvolený GSM modul Quectel M95. [21]	21
8	Znázornění hlavních komponent komunikátoru.	25
9	Schématické znázornění napájecích cest a napěťových úrovní.	26
10	Schéma zapojení obvodu pro ochranu akumulátoru.	27
11	Schéma zapojení obvodu pro nabíjení akumulátoru.	27
12	Schéma zapojení stabilizátoru na 3,3 V.	28
13	Schéma zapojení stabilizátoru na 4 V.	28
14	Schématické znázornění použitých komunikačních sběrnic.	29
15	Schéma zapojení FRAM paměti.	29
16	Schéma zapojení IQRF modulu.	30
17	Schéma zapojení napájecí větve IQRF modulu.	30
18	Schéma zapojení GSM modulu.	31
19	Schéma zapojení SIM konektoru.	32
20	Layout zařízení.	33
21	Sestavený akumulátor.	33
22	Výsledné zařízení.	34
23	Vyobrazení IPC.	36
24	Diagram aktivit chování komunikátoru.	43
25	Nastavení režimů DIP přepínačem.	44
26	Diagram aktivit měřicího cyklu.	45
27	Komunikační diagram měřicích úloh.	46
28	Komunikační diagram servisního režimu.	48
29	Diagram průběhu IQRF tasku vykonávající příkaz probuzení.	49
30	Diagram průběhu GSM tasku vykonávající příkaz probuzení.	50
31	Diagram průběhu GSM tasku vykonávající příkaz odeslání dat.	51
32	Schématické znázornění komunikace s datovým úložištěm.	53
33	Ukázka výstupu zpracovaných dat na straně serveru (vlevo servisní zpráva, vpravo datová).	56
34	Vývojový diagram zpracování HTTP PHP požadavku PHP skriptem.	57
35	Schématické znázornění servisního zapojení.	58
36	Naznačení servisní konfigurace IQRF MESH sítě. [15]	61

Seznam tabulek

1	Porovnání IQRF modulů. (duben 2019)	15
2	Mapa uspořádání F-RAM paměti.	47
3	Položky pro servisní konfiguraci komunikátoru.	60
4	Spotřeba elektrické energie.	62

Seznam výpisů zdrojového kódu

1	Struktura komunikačních zpráv IPC přijímaných systémovým taskem.	37
2	Struktura komunikačních zpráv IPC přijímaných IQRF taskem.	38
3	Struktura komunikačních zpráv IPC přijímaných GSM taskem.	38
4	Kódování zakázaných Byte.	52
5	Zakázané byte GSM modulu spolu s kódovacím znakem (encryptor).	52
6	HTTP request	53
7	Protokol servisní zprávy.	54
8	Parametry PHP skriptu pro zápis do souboru.	56
9	Protokol servisních příkazů pro konfiguraci komunikátoru.	58
10	Odpověď na přijatou servisní zprávu.	60
11	Příklad servisní komunikace.	61

1 Úvod

V dnešních dnech, které se označují jako čtvrtá průmyslová revoluce, si cestu do běžného života každého z nás nacházejí nejrůznější elektronická zařízení, která se snaží pokrýt všemožné oblasti lidské činnosti a zjednodušit či rozšířit sběr informací. Tento trend se nevyhýbá ani firemní sféře, kde se elektronika snaží nahrazovat a případně automatizovat činnosti, pro které není lidská síla efektivní. Tento celosvětový boom zahrnuje sféru zvanou Internet of Things (IoT) neboli počestně internet věcí. Pokud se bavíme o IoT můžeme si obecně představit nejrůznější malou a mnohdy i jednoduchou elektroniku využívající variace senzorů produkující data, jež jsou prezentována uživateli různými způsoby. Ať už potřebujeme kontrolovat klima budovy nebo exteriéru, ovládat osvětlení či hlídat přístupy do místnosti, nalezneme mnohá řešení jak za pomoci chytrých zařízení tyto informace získávat. Záleží na každém z nás, jak tato zařízení využijeme, zda se pokusíme sestavit vlastní systém nebo požádáme některou z firem o instalaci řešení pro námi požadované aplikace.

Ve světlech tohoto trendu není jednoduchá ani volba použitých zařízení. Ať už se jedná o kutilský projekt či firemní zakázku možnosti jsou široké, a vybrat ten správný ekosystém nemusí být triviálním úkolem. Roli v tomto směru hraje množství faktorů, kterými jsou například prostředí v němž se budou zařízení nacházet, dostupnost zdrojů energie pro napájení zařízení, rozlehlost systému, možnosti využití datových přenosů v závislosti na prostředí, zabezpečení a spolehlivosti, variabilita a škálovatelnost systému a mnoho dalších.

V souvislosti s tímto výběrem je také nutné bedlivě sledovat systémem podporované platformy a to v ohledech vizualizace dat či ovládání skrze mobilní přístroje a počítače. Tyto platformní volby navazují na možnosti podpory datových úložišť, která jsou dnes běžně dostupná jak v rozsáhlých variantách tak i ve variantách pro malé projekty s minimálními požadavky na financování. Jedná se především o populární cloudové služby, jejichž využití přináší mnohé výhody i nevýhody v závislosti na požadavcích na výsledek.

Samotný přenos dat od zařízení k datovému úložišti je dalším tématem, které hraje roli při počátečním výběru systému, ať už se jedná o využití známých a běžně dostupných informačních datových linek jako například Wi-Fi, Bluetooth, GSM, ethernet nebo speciálně vyvinutých linek, kterými mohou být například NarrowBand, LoRa, IQMESH a mnohé další.

Až samotná plánovaná oblast využití spolu se znalostmi různých IoT řešení tvoří základ pro vhodný výběr systému. V této práci je využito řešení firmy MICRORISC s.r.o., které pokrývá svou flexibilitou široké spektrum aplikací sahající od sběru dat v exteriérech po jednoduchá ovládací zařízení v uzavřených prostorách. Systém disponuje schopnostmi komunikace zařízení i bez přímé viditelnosti, především bez možnosti využití permanentních zdrojů energie, pro přenos menších objemů dat bez využití standardních komunikačních protokolů a bez využití kabelového spojení jednotlivých zařízení. Výsledkem je síť jednoduchých bezdrátově komunikujících zařízení za pomoci propracovaného komunikačního protokolu s nízkou spotřebou energie získávané primárně z akumulátorů.

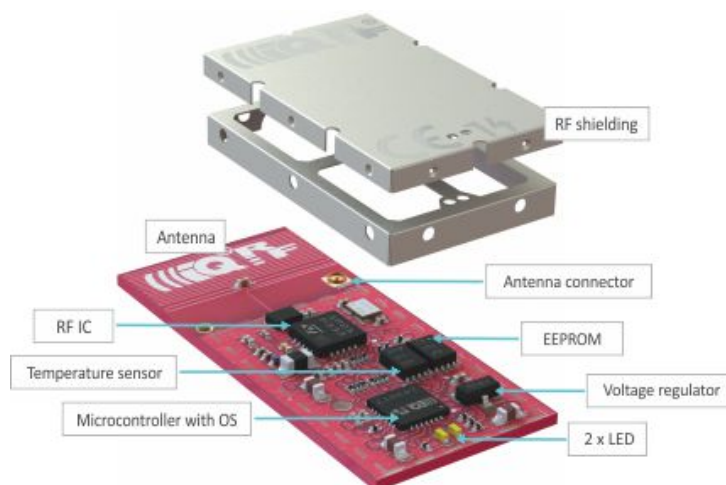
2 Popis použité technologie

2.1 IQRF

IQRF je technologie pro bezdrátový přenos. Zaměřuje se na přenos nízko-objemových dat s nízkou rychlostí pro užití v aplikacích s malou spotřebou elektrické energie. Tato technologie je výsledkem práce české firmy MICRORISC s.r.o., jež platformu IQRF založila roku 2004, a která je vlastníkem více jak dvou desítek patentů týkajících se jejich IQRF ekosystému. Platforma se již uplatňuje v oblastech monitorování, výrobní automatizace, v sítích senzorů pro exteriér i interiér, ve sportu a také v oblastech inteligentních budov.

2.1.1 IQRF hardware

Technologie pracuje na základě IQRF modulů, které tvoří hardware základ platformy. Jedná se o moduly malých rozměrů obsahující mikrokontrolér, EEPROM paměť, vstupně výstupní IO piny, analogové vstupy spolu s dalšími nezbytnými komponenty. Moduly obsahují vlastní operační systém, navíc svými rozměry mohou zapadnout do slotů SIM karet.



Obrázek 1: IQRF modul. [13]

Moduly jsou schopny komunikace mezi sebou a vytvoření tak bezdrátové sítě. Komunikace takovéto sítě probíhá na bezlicenčních rádiových frekvencích, konkrétně 868 MHz, 916 MHz nebo 433 MHz volba závisí na schopnostech vybraného modulu od této volby se také odvíjí možnost využít až 189 kanálů ve frekvenčním pásmu 916 MHz. Moduly jsou schopny vydat výstupního výkonu až 8 mW a pokrýt tak oblast rozsahu desítek metrů v budovách nebo stovek metrů ve otevřeném prostoru pouze s malými anténami. RF přenos probíhá s rychlostí 19,8 kb/s. Ve volbě modulů je možno vybírat z více typů, ty se mohou lišit v použitých frekvenčních pásmech, ve volbě použité antény nebo také v zakomponovaných periferiích.

Tabulka 1: Porovnání IQRF modulů. (duben 2019)

	TR-72D	TR-75D	TR-76D	TR-77D
Montáž	SIM slot	Vertical	SMT	SMT
Počet kontaktů	8	12	18	8
Počet I/O	6	10	12	4
MCU	PIC 16LF1938	PIC 16LF1938	PIC 16LF1938	PIC 16LF1938
Flash paměť	16 K x 14 b	16 K x 14 b	16 K x 14 b	16 K x 14 b
RAM paměť	1024 B	1024 B	1024 B	1024 B
EEPROM paměť	256 B	256 B	256 B	256 B
EEPROM ext.	32 KB	32 KB	32 KB	32 KB
RF IC	SPIRIT1	SPIRIT1	SPIRIT1	SPIRIT1
RF pásmo (MHz)	868/916/433	868/916	868/916/433	868
RF rychlost (kb/s)	19,8	19,8	19,8	19,8
RF výkon (max) (mW)	10	10	10	4
Proud deep sleep (nA)	1700	56	56	56
Proud sleep (uA)	2,3	0,61	0,61	0,61
Proud RX STD (mA)	12,1	11,8	11,8	11,8
Proud RX LP (uA)	260	250	250	250
Proud RX XLP (uA)	18,5	16,3	16,3	16,3
Proud TX (mA)	8,3 – 21,5	8,3 – 21,5	8,3 – 21,5	8,3 – 21,5
LDO	Ano	Ne	Ne	Ne
Napájecí napětí	3,1 – 5,3	3 - 3,4	3 – 3,4	3,1 – 3,4
A/D vstupy	2	3	3	3

Samozřejmostí je podpora několika komunikačních rozhraní jako SPI, UART nebo I²C a obousměrná komunikace modulů. Pro splnění nároků aplikací na nízkou energetickou náročnost mají moduly více režimů provozu, které při příjmu dosahují odběru 12 mA pro standardní režim, 250 uA pro low-power a 16,3 uA pro extra low-power režim. Odběr při odesílání je v rozsahu 8,3 mA až 21,5 mA, dále tyto moduly mohou být uspány a jejich odběr se pohybuje v řádech několika uA případně v módu hlubokého spánku se jedná o odběr až 56 nA.

Nabídka IQRF modulů je rozsáhlá a s vývojem se mění. Díky tomu, mohou být vyřazeny moduly, které již nabyly statusu "zastaralý", těmi jsou: TR-52D, TR-54D, TR-55D, TR-56D, TR-58D-RJ, TR-52B, TR-53B, TR-62D, TR-72D-WMB, TR-76D-WMB, TR-78D. V současné době tak k využití zůstávají čtyři typy modulů, jejichž porovnání vybraných parametrů uvádí tabulka 2.

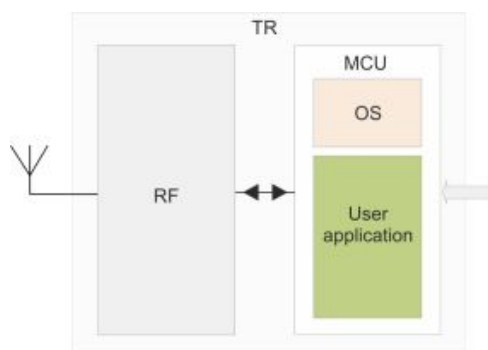
Z aktuální nabídky modulů dle tabulky byl pro tuto práci vybrán model TR-76D, a to především proto, že v porovnání s ostatními jsou spolu s modelem TR-77D jedinými moduly určenými

pro SMT montáž. Na rozdíl od TR-77D však disponuje vyšším maximálním RF výkonem, což je pro účel komunikátoru výhodné. Také jeho parametry v oblasti spotřeby elektrické energie jsou v porovnání s ostatními variantami nejlepší možné. Naopak díky SMT montáži neobsahuje vlastní napěťový regulátor a indikační LED, tyto součásti je tedy potřeba řešit externě mimo modul.

2.1.2 IQRF software

Aby bylo možno snadno implementovat moduly a omezit tak čas strávený vývojem, jsou moduly vybaveny softwarovou vrstvou představující funkcionalitu daného modulu. Moduly samy o sobě obsahují operační systém, který má předdefinovanou množinu funkcí, díky níž je usnadněn vývoj funkcionality.

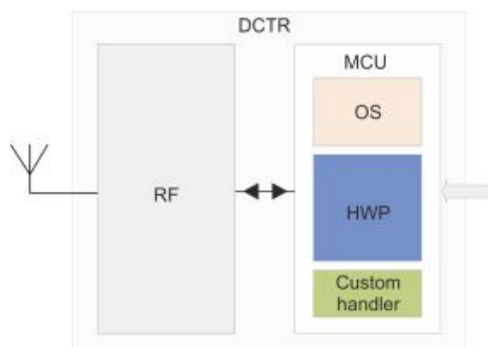
Existují dva přístupy, jak dosáhnout požadovaného chování IQRF modulu. První z přístupů využívá dvou softwarových vrstev, přičemž první je operační systém a druhou pak samotná uživatelská aplikace. Uživatelská aplikace, využívá funkcionalit OS, který je navržen konkrétně pro IQRF moduly, programuje se v jazyce C. Tento programátorský přístup je unikátní a patentovaný. Pro využití tohoto přístupu je zapotřebí znalost programování, navíc tato varianta nepodporuje síť IQMESH a tedy své využití nachází spíše v sítích peer-to-peer.



Obrázek 2: Struktura programování pod OS. [14]

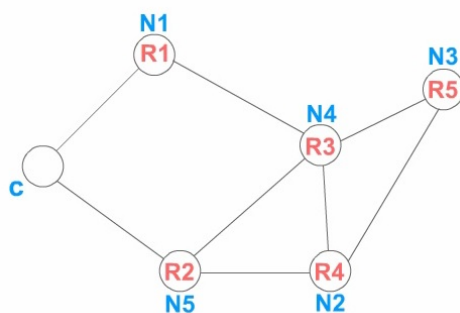
Existuje zde také možnost nastavení funkcionality modulů bez nutnosti programování. Ta je představována druhou variantou, která je složená ze tří softwarových vrstev využívajícího framework DPA. K původnímu OS zde přibývá vrstva hardwarových profilů HWP, což je softwarový plug-in, a případně volitelného handleru psaného v jazyce C. Opět se jedná o patentovaný koncept. Tato varianta podporuje IQMESH síť a její cíl je umožnění co nejsnazšího nastavení funkčnosti modulu bez nutnosti programování za pomoci grafického vývojového prostředí.

Přístup DPA umožňuje komunikaci modulů v IQMESH síti. Takováto síť je tvořena mesh topologií, v níž jsou datové pakety posílány mechanismem routování. Tento mechanismus umožňuje průchod dat skrze jiná zařízení v síti až k adresátovi. Díky tomu mohou být data doručena adresátovi i ze zařízení, která nemají přímý dosah. Datový paket tak od odesílatele prochází zařízeními, které je v dosahu, následně dalšími zařízeními až je doručen adresátovi. Díky využití



Obrázek 3: Struktura programování využívající DPA. [14]

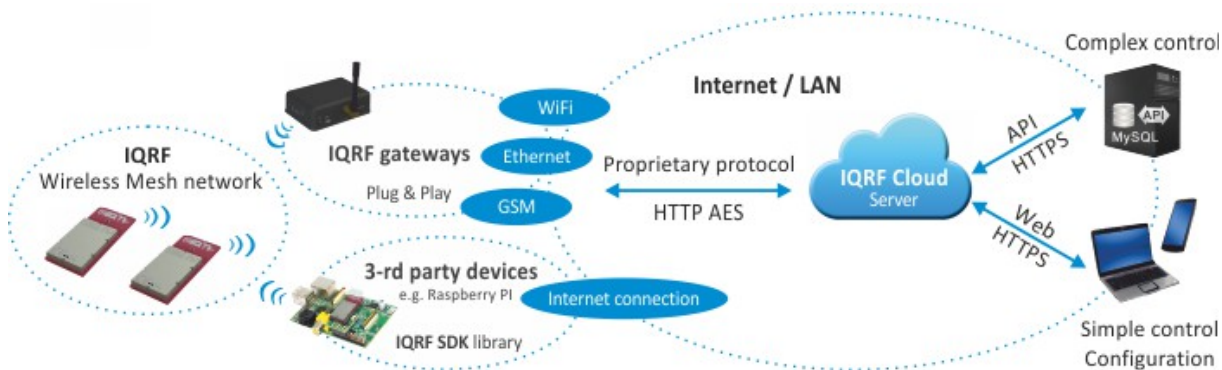
této topologie je možno dosáhnout daleko větších rozsahů pokrytí, což je nesmírnou výhodou ve celém spektru využití. IQRF využívá takovéto mesh sítě založené na protokolu IQMESH, který je součástí operačního systému, využívá se zařízení nazývaného koordinátor, které představuje centrální bod sítě. IQMESH síť disponuje možností propojení až 240 zařízení v síti, možností takzvaných skoků datových paketů po zařízeních, která nejsou adresáty, až do 240 skoků na paket v reálném čase. IQRF moduly v IQMESH síti mohou být nastaveny i jako běžné uzly sítě (node) i jako koordinátor, zároveň může každé zařízení pracovat ve dvou nezávislých sítích. IQMESH díky různým algoritmům routování podporuje například plnohodnotnou, redukovanou nebo optimalizovanou mesh síť, je schopna provádět automatické vyhledání cest mezi zařízeními nebo automatického vystavění sítě. Výhodou je pak i možnost nahrání aplikace do modulu bezdrátově.



Obrázek 4: Ukázka topologie mesh sítě. [5]

Pro snadnou prezentaci dat a komunikací s IQRF sítí je vybudován cloud server, který umožňuje zapojení IQRF zařízení do rozšiřující se sféry IoT. Komunikaci mezi samotnou IQMESH a cloudovým úložištěm zprostředkovává IQRF brána, která se stává účastníkem sítě IQRF a zároveň umožňuje komunikaci s internetem případně IQRF cloudem. Brány je možno vybírat v závislosti na požadavku spojení - GSM, Wi-Fi, Ethernet. Je možno využít také zařízení třetích stran, které budou v roli brány IQRF sítě, avšak zde již není podpora ze strany IQRF cloud. IQRF Cloud je přístupný za pomoci internetového prohlížeče a obsahuje veškeré informace při-

cházející z IQRF sítě, umožňuje grafickou vizualizaci, export dat, kontrolu sítě a také zasílání příkazů a parametrů nastavení do IQRF sítě. IQRF cloud podporuje připojení se pomocí platformy jako PHP, JavaScript a dalších, což může sloužit pro vytvoření vlastního úložného nebo prezentačního systému, který čerpá data prostřednictvím IQRF cloudu z IQRF sítě.



Obrázek 5: Struktura přenosu dat mezi IQMESH, IQRF cloud a uživatelskou aplikací. [6]

Práci s moduly v tomto ekosystému zajišťuje vývojový prostředek IQRF IDE, který je možno využít ve dvou variantách. První varianta je nástroj s grafickým rozhraním umožňující návrh, nahrávání, testování i údržbu prostřednictvím interaktivního projektově orientovaného prostředí. IQRF IDE využívá programátoru připojeného skrze USB a je navrženo pro OS Windows. Alternativou je nástroj příkazové řádky, který je zaměřen spíše na zkušené vývojáře, a umožňuje návrh efektivnějších a rychlejších návrhů. Zároveň je síla tohoto řešení v možnosti navrhnout funkcionalitu vývojářem a nechat již nezkušeného operátora nahrát tento program do zařízení bez nutnosti jakýchkoli dalších zásahů do programu. Operátor tak provede snadné nahrání již vytvořeného programu bez toho, aby do něj mohl zasáhnout, čímž je omezena možnost vzniku chyb. Opět je zde využito programátoru připojeného pomocí USB a IDE je vytvořeno pro OS Windows.

Aby bylo dosaženo plné kompatibility a jednoduchosti využití IQRF, jsou k dispozici také softwarové vývojové prostředky pro různé platformy. To umožňuje zapojení zařízení využívající Linux, Arduino, Raspberry Pi, BeagleBone nebo osobních PC. Díky takto rozsáhlé podpoře je IQRF snadno dostupné a tvoří plnohodnotný ekosystém.

2.2 NXP mikrokontrolér

Firma NXP, jakožto významný dodavatel procesorových zařízení ať už se jedná o 8, 16 nebo 32 bitové procesory, ale také dalších integrovaných obvodů, má ve svém sortimentu několik podmnožin mikrokontrolérů s různým využitím jak pro průmysl, tak pro koncové zákazníky. Těmito podmnožinami jsou:

- Aplikační procesory
- Procesory a mikrokontroléry založené na platformě ARM®

- Procesory a mikrokontroléry založené na architektuře Power Architecture®
- Mikrokontroléry a mikroprocesory pro automobilový průmysl
- Další procesory a mikrokontroléry

Ze široké škály procesorů nabízených firmou NXP byla zvolena řada Kinetis spadající do množiny procesorů a mikrokontrolérů stavěných na architektuře ARM®, jedná se o zařízení pro běžné vestavěné použití.

Řada Kinetis® L firmy NXP představuje sérii mikrokontrolérů cílených na IoT aplikace. Jedná se o 32-bitové MCU založené na jádru ARM® Cortex®-M0+, které kombinují nízkou energetickou náročnost spolu s rozsáhlými možnostmi volby periférií. Řada L obsahuje několik variant produktů, ty jsou řazeny do tříd KL0x, KL1x, KL2x, KL3x, KL4x, KL8x. Standardně obsahují všechny produkty řady Kinetis® L periférie UART, časovače, PWM, SPI, I²C, RTC a analogový komparátor. V čem se však produkty jednotlivých tříd liší jsou například frekvence jádra, velikost paměti, pouzdra zařízení a další zakomponované periférie. Základními kameny série L jsou:

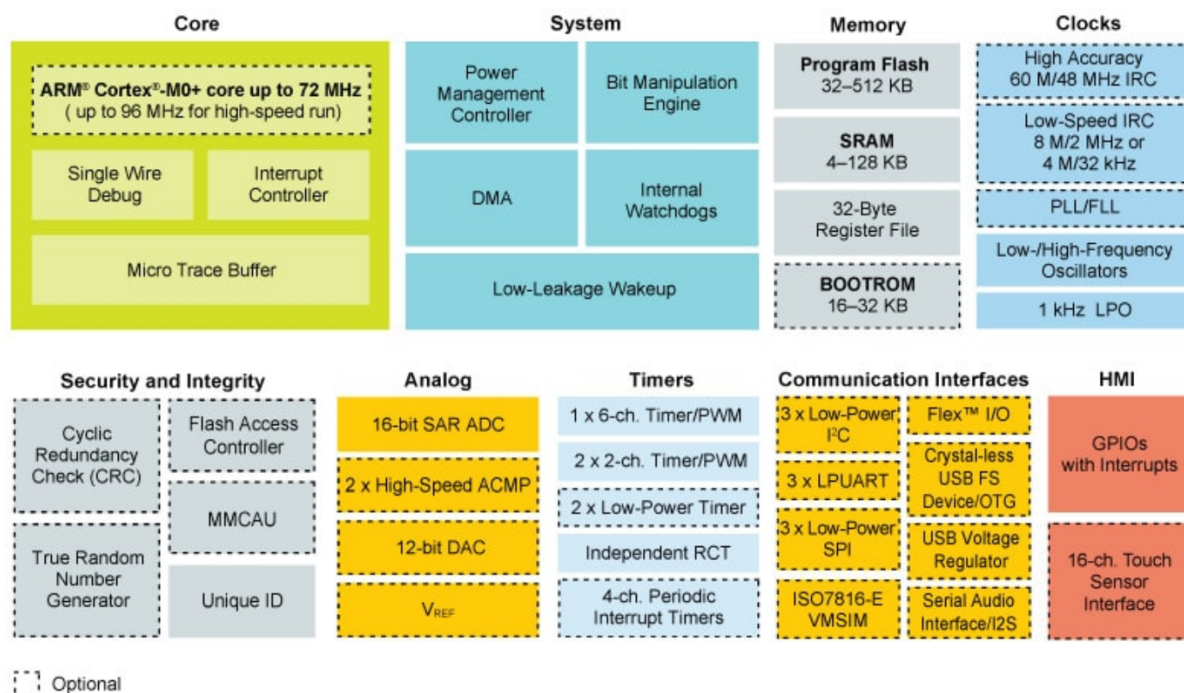
- Ultra-nízká spotřeba
- Ultra-malé rozměry
- Super snadné použití

Vybraný mikrokontrolér spadá do třídy KL2x. Tato třída je směřována k použití v oblastech herních ovladačů a příslušenství, PC periférií a příslušenství, přenosných zařízení a nositelné elektroniky. [19]

Mikrokontrolér, jež byl vybrán nese označení KL28 (MKL28Z512VLL7). Jedná se o variantu s 512 KB flash a 128 KB RAM, maximální frekvencí jádra 72 MHz ve 100pinovém obalu LQFP. Vybrané periférie KL28 [18]:

- 3x UART
- 3x SPI
- 3x I²C
- 1x I²S (audio)
- RTC
- ISO7816-3 (Identifikační karty)
- TSI (Touch Sensor Interface)
- FlexIO (Flexibilní IO emulátor rozhraní)

- 12-bit DAC
- 16-bit ADC
- 12-bit ADC
- 82x IO
- USB jako zařízení bez krystalu



Obrázek 6: Blokový diagram mikrokontrolérů série Kinetis L KL2x. [16]

2.3 GSM modul

Pro zajištění spojení datového komunikátoru s datovým úložištěm je potřeba zvolit vhodný prostředek pro komunikaci využívající GSM sítě v České republice. V této době se již široce využívají mobilní datové sítě třetí generace, jež jsou na území České republiky provozovány všemi hlavními operátory. Avšak pro účely této práce není nutné využívat sítí s takovými parametry jako mají například sítě LTE. Jelikož se mesh sítě IQRF využívají pro sběr poměrně malých dat není nutné klást velké nároky na GSM modul, který bude zajišťovat přenos dat, v oblasti přenosových rychlostí.

Modul byl vybrán ze sortimentu firmy Quectel, která je globálním dodavatelem průmyslových celulárních a GNSS modulů, ty jsou určeny pro široké aplikační využití - jak pro IoT, tak také automobilový průmysl, smart zaměření, dálkové ovládání, měření a sledování, a pro zabezpečení

a ochranu zdraví. Portfolio firmy se zaměřuje na moduly LTE, LTE-A, LPWA, UMTS/HSPA(+), GSM/GPRS a GNSS.

Konkrétně zvolený model je modul Quectel M95, jenž je schopen pracovat s datovou sítí GPRS (2G) a zvládá využívat protokoly TCP, UDP, HTTP, FTP, PPP, SMTP a další. Modul disponuje nízkou spotřebou elektrické energie, což je pro toto použití klíčové spolu se základními schopnostmi komunikace po datové síti. Díky podporovaným frekvencím 850, 950, 1800 a 1900 MHz je vhodný pro podmínky pokrytí mobilní sítí v České republice. Dalšími výhodami zvoleného modulu jsou dobrá pážitelnost pouzdra a přívětivá tržní cena. To vše z modulu M95 dělá vhodnou volbu pro potřeby této práce.



Obrázek 7: Zvolený GSM modul Quectel M95. [21]

2.4 Zdroj energie

Standardně dostupné zařízení obdobného účelu využívají ke své funkci především energii z permanentně dostupných zdrojů energie - síťových rozvodů. Tato práce jde v ohledu volby napájení jiným směrem a tím je, v korespondenci s požadavkem na nízkou spotřebu elektrické energie, napájení z akumulátoru. Volba takového akumulátoru musí přihlížet k některým požadavkům na zařízení. Jsou jimi například vhodné napájecí napětí, fyzická velikost, schopnost dodávat dostatečný proud, kapacita akumulátoru pro zajištění provozuschopnosti zařízení po dostatečnou dobu. Díky poměrně aktivnímu rozvoji v oblasti hledání prostředků pro uchovávání elektrické energie, máme dnes široký výběr dostupných a kvalitních akumulátorů. Běžně je možno použít pro napájení akumulátory různých technologií, které s sebou přinášejí soubor vlastností, na jejichž základě je zvolen vhodný akumulátor pro aplikaci.

První technologií jsou akumulátory olovené, jejichž pozitivní vlastností je především nízká prodejní cena, jelikož se jedná o poměrně zaběhlou a běžnou technologii. Použití těchto akumulátorů je dnes zcela běžné a každý se může setkat s těmito zdroji elektrické energie minimálně u

automobilů. Akumulátory této technologie se standardně dodávají s nominálním napětím 12 V, jsou poměrně těžké, rozměrné a za nízkých teplot ztrácejí kapacitu. Výhodou však je schopnost dodávat velké hodnoty proudu, nízké samo-vybíjení a jednoduché dobíjení.

Další technologií, která zde stojí za zmínku, je Li-Ion. V dnešní době se jedná o poměrně široce používanou technologii akumulátorů pro elektroniku. Články těchto akumulátorů nesou nominální napětí o hodnotě 3,6 V až 3,7 V, nabízejí vysoký poměr hustoty energie na objem, a díky tomu jsou akumulátory menších rozměrů při dodržení obdobné kapacity v porovnání s olovenými akumulátory. Navíc jsou tyto akumulátory značně lehčí. Technologie již někdy nahrazuje olovené akumulátory v motocyklech a zahradní technice. Velikou výhodou zde je nízká hodnota samo-vybíjení, naopak jsou tyto akumulátory náchylné na překročení limitních hodnot při nabíjení i vybíjení, kdy je možné akumulátory zničit.

Technologie Li-Pol, která vzešla z velice úspěšné Li-Ion, je také k dispozici v široké nabídce. Tyto akumulátory mají shodné nominální napětí článků jako Li-Ion, navíc je tato technologie obecně vyspělejší a nabízí tak lepší hodnoty většiny parametrů jako jsou samo-vybíjení, schopnost dodat vysoké proudy, poměr energetické hustoty na objem a zvětšený počet pracovních cyklů. Nevýhody Li-Ion přetrvávají a přidává se k nim náročnější výroba, která se projevuje na vyšších cenách těchto akumulátorů.

V poslední řadě je potřeba zmínit také technologie niklových akumulátorů. Jsou jimi Ni-MH, Ni-Cd, Ni-Zn. Tyto pracují s nižšími nominálními napětími v rozsahu 1,2 V až 1,5 V. Jednotlivé technologie se liší především ve schopnostech pracovat do vysokého vybití, trpí ale vyšším samo-vybíjením. V porovnání s novějšími technologiemi Li-Ion a Li-Pol mají nižší poměr energetické hustoty na objem.

Pro tuto práci byl zvolen akumulátor technologie Li-Ion, jakožto zástupce moderní technologie akumulátorů s ohledem na pořizovací náklady a dostupnost. Technologie vyhovuje také díky svému nominálnímu napětí a vysoké energetické hustotě, která umožňuje menší rozměry, což vede k celkové rozměrové úspoře zařízení.

3 Možnosti přenosu dat ze sítě IQRF na datové úložiště

Jak již bylo zmíněno v kapitole 2.1.2, existuje IQRF cloudové úložiště komunikující s IQRF bránami. Jelikož se tato práce zabývá právě návrhem datového komunikátoru pro IQRF síť, bylo by vhodné využít právě stávající IQRF cloud a doplnit tak již funkční ekosystém. IQRF Cloud je produktem firmy CIS, s.r.o., která patří do IQRF Alliance. IQRF brány s cloudovým úložištěm komunikují daným protokolem. Na základě proběhnuvší komunikace s firmou CIS, s.r.o. bylo zavrženo možné využití IQRF cloud, jelikož protokol používaný pro komunikaci s IQRF bránami je neveřejný, tudíž se nelze tímto směrem vydat. Firma CIS, s.r.o. do budoucna plánuje implementaci MQTT kanálu pro účely propojení bran třetích stran, do té doby se jedná o uzavřenou službu.

Protokol MQTT (Message Queuing Telemetry Transport) je náhražkou dříve uvažovaného přenosu dat protokolem HTTP. Potřeba přejít na MQTT protokol vznikla z náročnosti HTTP protokolu. MQTT je stručnější protokol, který využívá TCP, obsahuje minimum servisních dat a je tak snadněji realizovatelný na zařízeních s malým výpočetním výkonem - tudíž je ideální pro použití v IoT. Protokol neudává jaká data se přenáší, jeho využití je tak všestranné.

3.1 Veřejně dostupná úložiště

Díky možnostem, které nabízí GSM moduly je reálně možno se připojit k libovolnému serveru na internetu. To přináší svobodu v rámci výběru cloudových úložišť. S rozšiřujícím se IoT roste také množství veřejně dostupných úložišť nabízených různými poskytovateli. Mezi nejviditelnější patří například Amazon AWS IoT, Microsoft Azure, IBM Watson IoT, Google Cloud a mnoho dalších.

Amazon AWS IoT Cloud využívá právě již zmíněný MQTT protokol, HTTP a také Web-Sockets. Navíc umožňuje komunikace mezi zařízeními nezávisle na využitém protokolu. Využívá ověřování identity zařízení, čímž zajišťuje bezpečnost. Amazon doplňuje AWS Cloud o služby umožňující využití uložených dat, jejich filtrování, vizualizaci, přenos a mnoho dalšího. Také obsahuje nástroje pro konfiguraci připojených zařízení, parametry zařízení jsou uloženy na cloudu a mohou být kdykoli změněny, cloud se již postará, aby byly po připojení zařízení tyto parametry přeneseny a reálně nastaveny. Služba je účtována částkou 5 dolarů za každý milión zpráv. Zpráva se zde rozumí blokem 512 bytů.

Azure je cloudové úložiště z dílen Microsoft, které se pyšní řadou dodržovaných předpisů, uznáním jako nejdůvěryhodnější cloud pro americké orgány státní správy a autorizací FedRAMP. Nabízí možnost hybridního užití - spojením cloudových služeb a aplikací pro PC v úzkém propojení. Azure uvádí, že má největší množství datových center po celém světě, navíc podporuje opensource projekty a je partnerem Red Hat, což vede na možnost využití tohoto řešení pro různé platformy, různé vývojářské systémy. Zabezpečení je řešeno podobě jako u Amazon AWS pomocí identit jednotlivých zařízení, navíc má možnost využití nástroje pro hromadnou registraci zařízení, využití umělé inteligence a jiné pokročilé analýzy. Cena za využívání základní

služby Microsoft Azure IoT Centrum se liší v závislosti na zvoleném tarifu, který se odvíjí od množství zpráv za den. V rozsahu do 8000 zpráv za den, přičemž je jako zpráva chápán blok 0,5 kB je využívání služby zdarma. Následují placené tarify, jejichž zprávy jsou chápány jako blok o 4 kB, zde se jedná o 400 000 zpráv, 6 000 000 zpráv a 300 000 000 zpráv denně s cenami 50, 500 a 5000 dolarů měsíčně.

Google nabízí podporu svého IoT úložiště pro širokou množinu vestavěných operačních systémů, Android Things a také poskytuje podporu pro zařízení například od společností Intel nebo Microchip. Zabezpečení nabízí asymetrickou autorizaci díky TLS 1.2, případně pomocí certifikátů. Tak jako předchozí poskytovatelé i Google nabídne množství aplikací doplňující IoT cloud o možnost publikace a zpracování dat. Komunikace se zařízeními může využívat MQTT, HTTP protokoly. V případě Google IoT Core se zatím jedná o beta vydání. Ceny jsou účtovány dle objemu přenesených dat, přičemž jako minimální velikost zprávy je účtováno vždy 1024 B, přestože reálná velikost zprávy může být menší. Objem dat do 250 MB je zdarma, do 250 GB je cena 0,0045 dolarů za MB, do 5 TB je cena 0,002 dolarů za MB a nad tento limit je cena 0,00045 dolarů za MB.

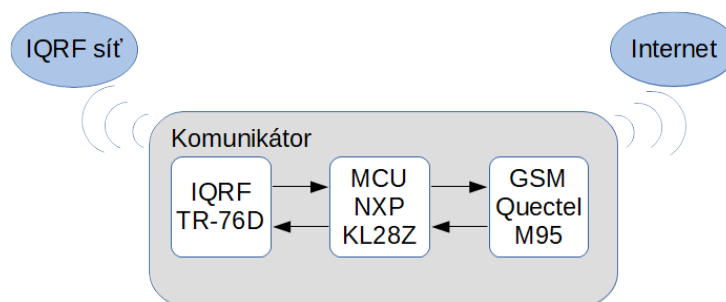
IBM Watson IoT nabízí snadnou a bezpečnou registraci a připojení zařízení s podporou instruktážních návodů krok-za-krokem pro populární zařízení - senzory i brány. Zákazníkům je umožněna analýza dat v reálném čase - pozorování toku dat nastavenými pravidly/filtry, analýzami spolu s definováním pravidel pro spouštění událostí a alarmů umožňujících zasílání e-mailů a dalších akcí pro reakci na kritické změny dat. IBM nabízí tři tarify, z nichž první je zdarma, jedná se o tarif do 500 registrovaných zařízení a maximálním objemem dat 200 MB na přenos. Tento tarif zahrnuje možnost analýzy dat a tzv. edge analýzy, což je možnost spouštět analýzu dat mimo cloudové úložiště - na externím HW tam, kde je to pro data smysluplné. Další tarif je nadstavbou ve smyslu využití datového toku, kdy je za přenos dat účtováno po MB dle přenosu, navíc dle analýzy na MB plus dle edge analýzy na MB. Cena je tak složena z uvedených tří složek. Poslední tarif nabízí rozšíření o management bezpečnosti a rizik, ceny jsou účtovány obdobně jako u předchozího tarifu.

3.2 Použité úložiště

Nabídka provozovatelů cloudových úložišť s podporou IoT je široká a zároveň existuje spousta podpůrných prostředků pro zpracování dat, které se liší cloud od cloudu. V nabídce a použitelnosti těchto nástrojů je nesnadné se orientovat, zároveň může být náročným úkolem záměna využívaného úložiště. Bohužel nelze ani využít IQRF Cloud a doplnit tak nabídku IQRF bran v plné použitelnosti. Na základě těchto faktů nebude pro potřeby této práce využito cloudového úložiště, ale webového serveru, což je volně dostupná služba, kterou je možné zřídit také zdarma. V rámci komunikace s běžnými webovými servery je využito protokolu HTTP, který v porovnání s MQTT sice není uzpůsoben pro aplikace sběru dat jako je tato, avšak jedná se o zcela běžně podporovaný protokol a není tak pochyb o kompatibilitě s libovolným datovým úložištěm.

4 Návrh a realizace hardware komunikátoru

Na základě kapitoly 2 byly vybrány tři stěžejní hardwarové komponenty tvořící páteř komunikátoru. Tato kapitola rozebírá jejich konkrétní hardware zapojení. Schéma a s nimi spojený návrh desky plošných spojů (layout) byly vytvořeny za využití software Autodesk EAGLE verze 9. Výsledné schéma s návrhem desky plošných spojů jsou přiloženy v elektronické podobě k této práci. Pro vytvoření kompletního hardware bylo využito více jak 200 kusů různých elektronických součástek, které jsou vyhledatelné právě ve schématu nebo také v sestaveném seznamu použitých součástek (BOM), který je také přiložen v elektronické podobě. V této kapitole jsou zmíněny pouze hlavní funkční celky, které jsou stěžejní.

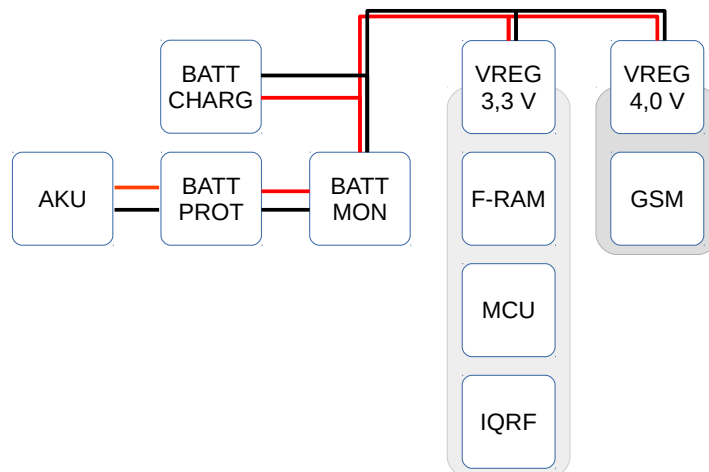


Obrázek 8: Znázornění hlavních komponent komunikátoru.

4.1 Napájení

Jelikož se předpokládá využití GSM-IQRF komunikátoru v exteriéru, bylo jako zdroj napájení zvoleno zapojení akumulátorů. Konkrétně Li-Ion bateriových článků, jejichž nominální napětí je 3,7 V a kapacita 3200 mAh. GSM modul je dle datasheetu výrobce napájen typickou hodnotou napětí rovnou 4 V což je hodnota vyšší než článek akumulátoru poskytuje. Pro zajištění dostatečně vysokého napětí pro napájení GSM modulu se nabízí více variant. Možnostmi jsou například využití DC-DC měniče nebo sériového propojení více článků. V případě této práce bylo zvoleno sériové zapojení dvou článků akumulátorů, tvořící výsledné napájecí napětí $2 \times 3,7 \text{ V}$ tedy 7,4 V. Pro zvýšení kapacity a tudíž i doby provozu zařízení mezi dobíáním akumulátorů byly tyto dva sériové články zdvojeny také paralelně. Výsledně je tedy napájení tvořeno čtyřmi Li-Ion články dodávající napětí 7,4 V a kapacitu 6400 mAh. Články jsou typu LG MH1, konkrétně GWL Power GB18650.

Li-Ion akumulátory budou během svého provozu energii dodávat, ale také přijímat během nabíjecích a vybíjecích cyklů. Je nutné akumulátory udržovat v kondici pro zajištění jejich trvanlivosti a také zajistit pracovní podmínky, které nepovedou k nenávratnému poškození akumulátorů. Pro tyto účely byly vytvořeny integrované obvody, které jsou také v této práci využity. Jsou jimi BQ29209 firmy Texas Instruments, což je napěťová ochrana druhého stupně s automatickým balancováním pro dva sériové články a obvod pro ochranu prvního stupně BQ28Z610 také od

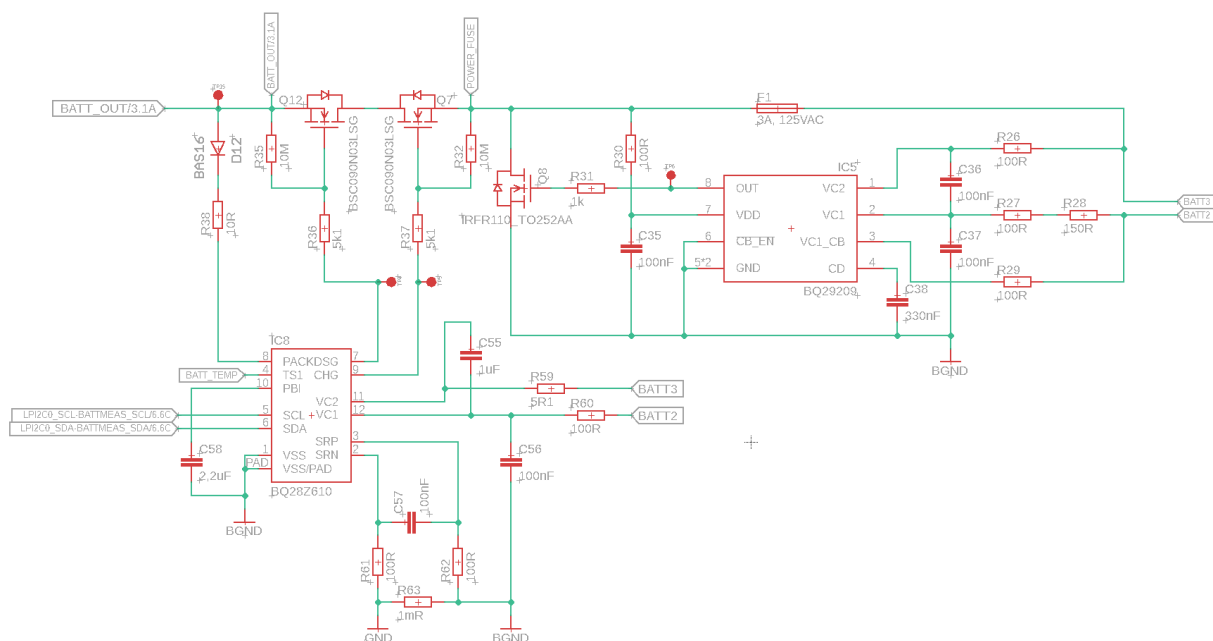


Obrázek 9: Schématické znázornění napájecích cest a napěťových úrovní.

Texas Instruments. První zmíněný integrovaný obvod má za úkol balancovat sériově zapojené články a chránit je proti přepětí. V případě nutnosti je schopen přepálit tavnou pojistku, která je umístěna v osazeném lóže na desce plošných spojů, a ochránit tak akumulátory například před možným vznícením či explozí. Druhý zmíněný je schopen chránit akumulátory proti podpětí - tedy vybití pod únosnou hodnotu napětí, dále například proti přehřátí, přepětí a také proti nad-proudu. Jde o první stupeň ochrany a tento obvod již nedisponuje možností odpojení akumulátorů radikální cestou spálením tavné pojistky, ale pouze uzavřením dvojice tranzistorů, které se nacházejí v napájecí cestě. Parametry napěťových a proudových mezí je možno pro první stupeň ochrany nastavovat díky komunikační sběrnici I²C, která také umožňuje využívat integrovaný obvod pro zjišťování parametrů připojených akumulátorů jakými jsou například aktuální napětí, dodávaný proud, životnost a mnoho dalších parametrů. Této monitorovací funkcionality je v práci využito a sběrnice prvního stupně ochrany je přivedena do MCU. Pro první nastavení bez možnosti využít MCU je vyveden konektor na tuto sběrnici I²C. Obvod umožňuje měření parametrů akumulátorů na základě rezistoru o hodnotě 1 mΩ spojující zemi cesty hardware komponent se záporným pólem akumulátoru. Před použitím obvodu je však nutné nastavit jej na základě proudové a napěťové kalibrace a nastavením chemického složení pro konkrétní články. Tyto úkony byly provedeny za pomoci vývojového prostředku Texas Instruments EV2300 EVM Interface Board a Battery Management Studio (bqStudio) software.

Zapojení obvodů pro ochranu akumulátorů, ve schématu 10 označených jako BATT2 a BATT3 pro zapojení dvou článků sériově, bylo vytvořeno podle pokynů vytvořených výrobcem v dokumentech k těmto obvodům. Dle maximálního špičkového odběru GSM modulu byla zvolena hodnota proudu tavné pojistky na 3 A včetně tranzistoru Q8, který musí být schopen podle pokynu obvodu ochrany druhého stupně způsobit spálení tavné pojistky bez sebe-poškození.

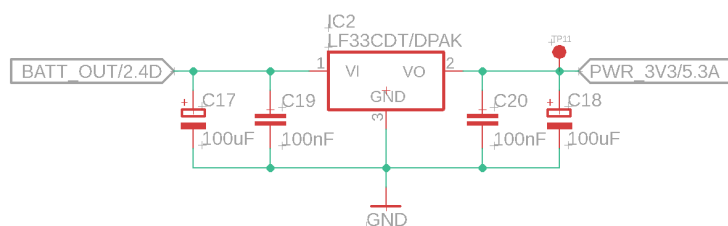
Pro nabíjení akumulátorů byl vybrán integrovaný obvod pro management nabíjení s ochranou proti přepětí Microchip MCP73213. Vstupní napětí může být maximálně 16 VDC a obvodem



maximální dodávaný proud do akumulátorů je 1100 mA. Ochrana před připojením vstupního napájení, které by mohlo poškodit obvody je ve formě průmyslového konektoru napájení - není tak možno připojit zdroj napětí standardních konektorů. Během napájení jsou stále aktivní oba ochranné prvky akumulátoru.

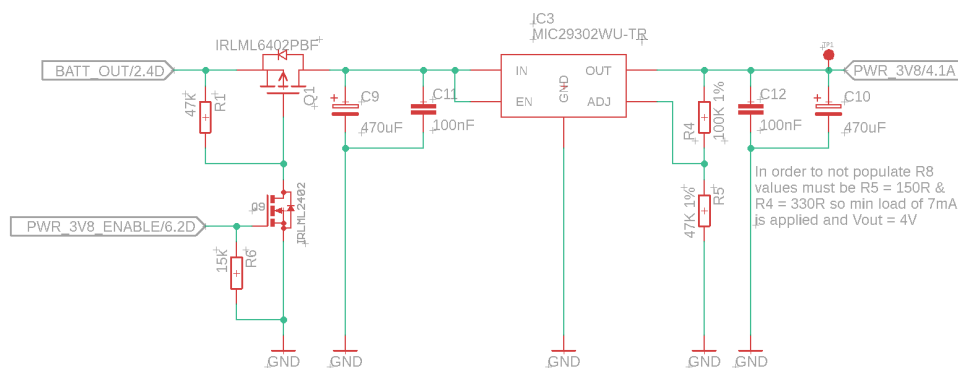
Pro napájení jednotlivých komponent zařízení je nutno zajistit stabilní napěťové úrovně dle požadavků daných komponent. Je zapotřebí dvou napájecích úrovní - standardní 3,3 V pro MCU a IQRF modul a navíc 4 V pro GSM modul. Pro tyto účely bylo použito dvou stabilizátorů stejnosměrných napětí a jejich zapojení dle doporučení výrobců.

Stabilizátor 3,3 V pro napájení MCU a IQRF modulu je schopen dodávat až 500 mA, a jedná se o LF33CDT firmy STMicroelectronics. Tento stabilizátor pevného napětí je připojen přes ochrany na akumulátor s nominálním napětím 7,4 V a není možné jej vypnout a vyřadit tak hlavní komponentu zařízení - MCU.



Obrázek 12: Schéma zapojení stabilizátoru na 3,3 V.

Stabilizátor na 4 V pro napájení GSM modulu je nastavitelný stabilizátor napětí, jehož výstupní napětí je dáno napěťovým děličem na pinu ADJ. Stabilizátor umožňuje povolení nebo zakázání výstupu pomocí jednoho ze vstupních pinů, což v této práci využito není a stabilizátor dodává na výstupu nastavenou napěťovou úroveň kdykoli je napájen. Avšak v rámci úspory elektrické energie je stabilizátoru předřazen tranzistor, jehož účelem je za pomoci MCU odstavit tento stabilizátor z provozu přerušením napájecí cesty na vstupu stabilizátoru. Je využit stabilizátor MIC29302WU-TR od Microchip schopný dodávat až 3 A, což pokrývá katalogem uváděnou špičkovou spotřebu GSM modulu.

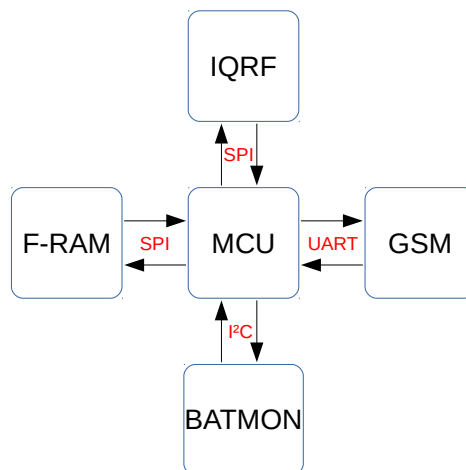


Obrázek 13: Schéma zapojení stabilizátoru na 4 V.

4.2 Zapojení MCU

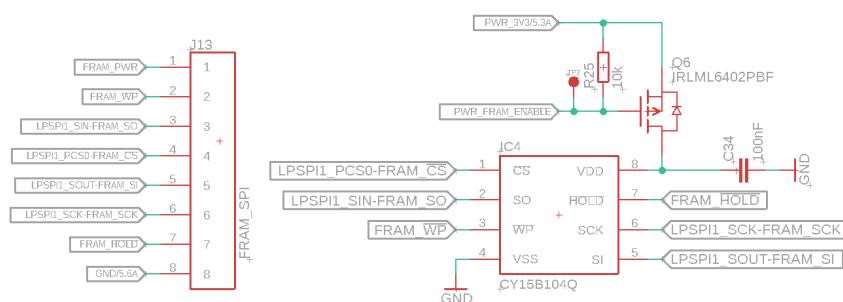
Mikrokontrolér KL28Z je připojen na napájení 3,3 V, v jeho blízkosti je umístěno zapojení blokovacích kondenzátorů, na referenční vstup a analogové napájení je připojeno zapojení pasivních LC filtrů.

Zapojení mikrokontroléru disponuje celkem třemi rozhraními/konektory pro vnější komunikaci. Jsou jimi SWD konektor sloužící pro ladění a nahrávání firmware, USB konektor pro servisní komunikaci s PC a třetím je konektor sloužící pro externí napojení na sběrnici I²C o napěťové úrovni 3,3 V. Tato sběrnice s konektorem slouží pro nastavení bateriového monitoru (ochrany prvního stupně) externím zařízením.



Obrázek 14: Schématické znázornění použitých komunikačních sběrnic.

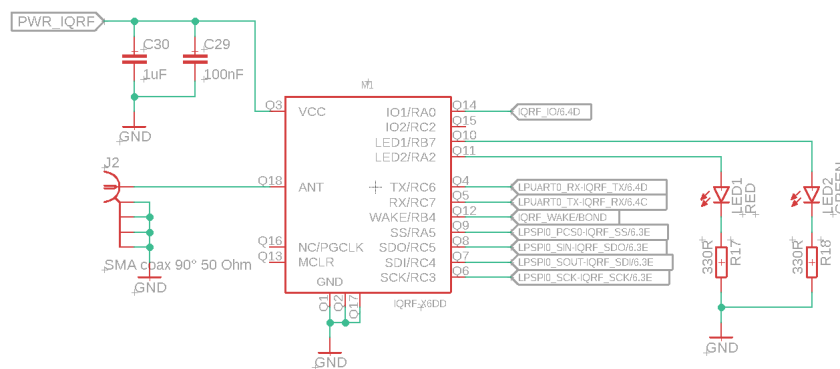
Zbylé využití piny MCU slouží k připojení a komunikaci s ostatními periferiemi za pomoci sběrnic UART, SPI, I²C a digitálních vstupů/výstupů. MCU je napojen navíc na externí krystal o frekvenci 32 768 KHz pro funkcionalitu hodin reálného času. Pro možnost uložení dat a nastavení je zapojení doplněno o F-RAM paměť, která je připojena sběrnicí SPI. Zároveň je napájecí větev F-RAM připojena na MCU tak, aby bylo možno paměť odpojit od napájení, přičemž tato paměť také využívá napájecí napětové úrovně 3,3 V. Komunikační sběrnice SPI sloužící k přístupu do paměti je vyvedena na konektor, který není osazen a slouží k usnadnění vývojových prací při tvorbě firmware. K MCU jsou připojeny celkem čtyři LED, jedno reset tlačítko a čtyř-bitový DIP přepínač. DIP přepínač je využit k výběru funkčních režimů zařízení. Tlačítko reset je zapojeno navíc s jedním DIP přepínačem pro možnost trvalého sepnutí reset pro MCU. K MCU jsou připojeny také čtyři spínací tlačítka, která však nejsou využita a díky tomu nejsou ani na desce plošných spojů osazena.



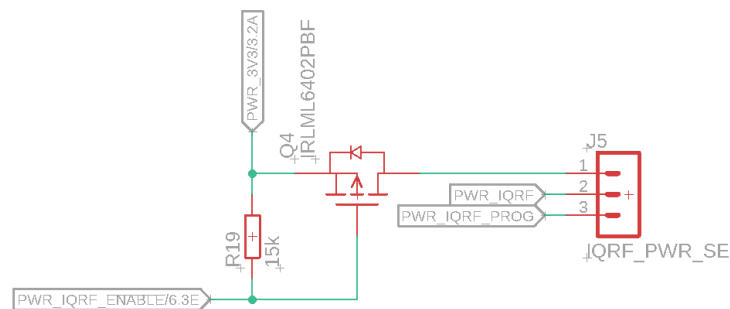
Obrázek 15: Schéma zapojení FRAM paměti.

4.3 Zapojení IQRF modulu

IQRF modul je napájen stejným napětím jako MCU, tedy 3,3 V ze stejného napěťového regulátoru. Modul je vybaven celkem dvěma konektory, přičemž jeden slouží jako programovací, druhý jako přepínač zdroje napájení. Zdroj napájení je možno zvolit konektorem pomocí jumperu mezi napájením z akumulátorů a napájením skrze programovací konektor z externího programátoru. Také napájení IQRF modulu, konkrétně jeho napájecí větev napojenou na akumulátor skrze napěťový stabilizátor, nikoli externí napájení, je možno programově vypínat z MCU. Na IQRF modul jsou připojeny dvě LED signalizující jeho činnost, koaxiální konektor pro připojení antény. IQRF modul je přímo spojen s MCU pomocí sběrnic SPI, UART a dvou digitálních signálů - WAKE/BOND a IO.



Obrázek 16: Schéma zapojení IQRF modulu.



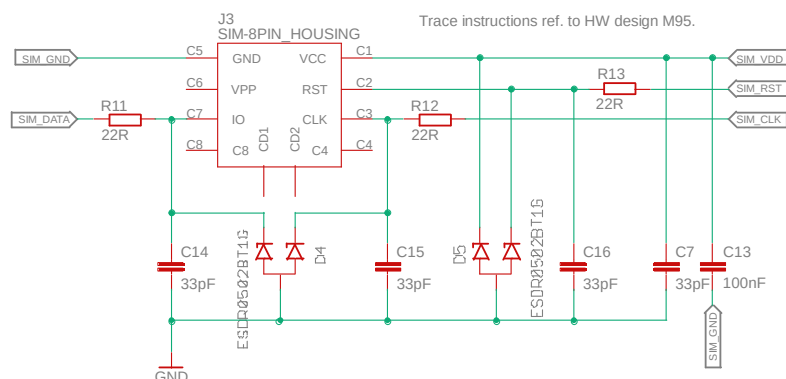
Obrázek 17: Schéma zapojení napájecí větve IQRF modulu.

4.4 Zapojení GSM modulu

Jak již bylo zmíněno Quectel M95 pracuje s nominálním napětím 4 V, je tedy napájeno z jiného regulátoru napětí než tomu bylo u MCU a IQRF modulu. Zapojení kondenzátorů na vstupních pinech napájení GSM modulu spolu s minimální šířkou napájecí cesty odpovídá doporučení výrobce.

[illegible]

31

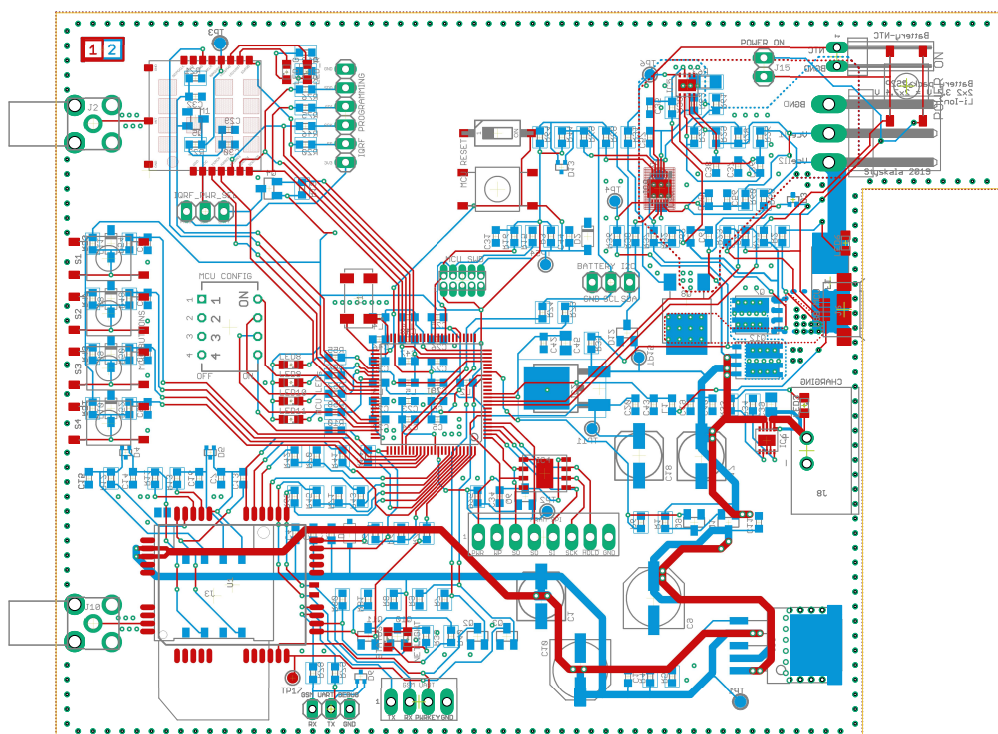


Obrázek 19: Schéma zapojení SIM konektoru.

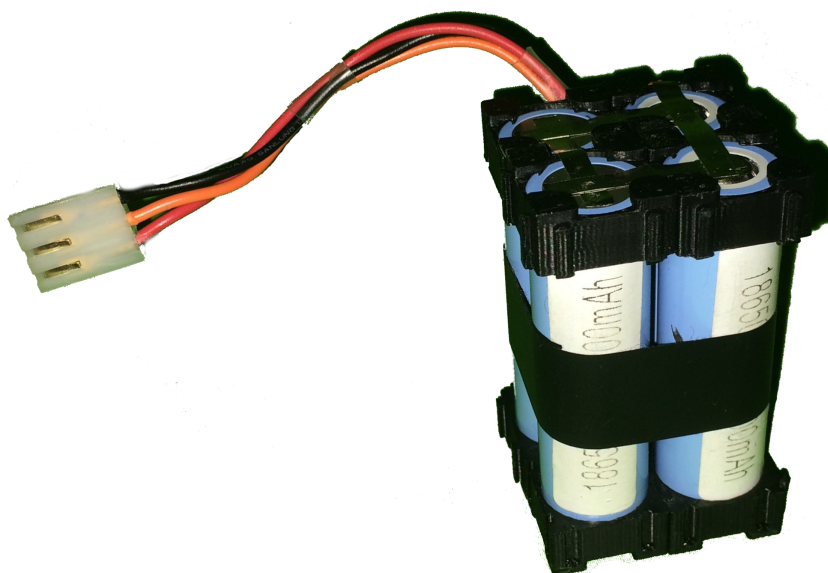
4.5 Výsledné zařízení

Použitím zapojení uvedených v předchozích kapitolách spolu s jejich vzájemným propojením a dalšími méně významnými obvody vzniklo zařízení tvořící výsledný komunikátor. Toto je složeno z jedné desky plošných spojů ve dvouvrstvém provedení se silou mědi $35\ \mu\text{m}$ a jádrem o tloušťce 1,6 mm. Bylo využito součástek typu SMD, které byly osazeny v rámci této práce svépomocí. Tvar desky plošných spojů byl navržen s ohledem na umístění v krabici spolu s akumulátorem. Layout desky je vyobrazen na obrázku 20, kde není vyobrazeno plošné vylití země kvůli lepší čitelnosti.

Deska je pro pracovní využití umístěna v hliníkové krabici spolu s akumulátorem, který byl pomocí odporové svářečky sestaven ze čtyř samostatných článků. Krabička má plastová čela, přičemž jedním z čel procházejí konektory pro připojení antén pro GSM a IQRF moduly.



Obrázek 20: Layout zařízení.



Obrázek 21: Sestavený akumulátor.



Obrázek 22: Výsledné zařízení.

5 Návrh a realizace firmware

Samotný hardware komunikátoru by pro funkcionalitu nestačil, v rámci práce tak byl vytvořen firmware, tedy program díky kterému teprve zařízení provádí programem definovanou činnost. Jedná se patrně o nejnáročnější a nejrozsáhlejší část této práce, jelikož sahá od nastavování registrů MCU až po sestavování HTTP požadavku, přičemž umožňuje jistou míru konfigurace prostřednictvím čtyř-bitového DIP přepínače pro provozní režimy, a prostřednictvím servisní komunikace editovatelné chování v sítích IQRF a GSM. Samotný mikrokontrolér KL28Z příliš firmware nezjednodušuje, protože se jedná o MCU splňující vysoké nároky na flexibilitu a také jeho periferie jsou vysoce škálovatelné nejen v rámci snižování energetické spotřeby. Jako příklad může posloužit samotné nastavení hodin pohánějící komunikační periferie, které mohou být zvoleny až ze čtyř alternativ, které nabízejí navíc nastavitelnou děličku, která může být dále dělena děličkami jednotlivých periférií. To umožňuje zvolit optimální hodiny každé periférie v rámci potřebné frekvence pro její činnost a zároveň s ohledem na spotřebu energie. Současně však je potřeba brát ohled na ostatní periferie využívající stejný zdroj hodinového signálu.

V rámci usnadnění vývoje na svých produktech i firma NXP nabízí volně k dispozici SDK podporující také KL28Z. Po seznámení se s tímto prostředkem a několika zkouškách jeho použití byl zvolen přístup vývoje firmware bez využití SDK. Důvody tohoto rozhodnutí byly především jeho velká návaznost na různé FW moduly, které nejsou v této aplikaci potřebné a využité, dále díky těmto návaznostem komplikované uvedení do provozu a v poslední řadě nedostatečná dokumentace, jež místy neodpovídala skutečnému obsahu SDK. V rámci této práce tak vznikly programové celky obsluhující použité periferie v návaznosti na požadavky zařízení k nim připojených. Tyto celky jsou v práci nazývány jako ovladače a příkladem může být ovladač pro LPSPI periférii komunikující s IQRF modulem, který obsahuje specifické potřeby časování a přenosů dat vyplývající z požadavků IQRF modulu. Přestože periférii LPSPI využívá také energeticky nezávislá paměť F-RAM, ovladač k periférii pro použití s tímto externím čipem je vytvořen odlišně a odděleně od ovladače LPSPI pro IQRF modul, jelikož požadavky na časování a průběh komunikace se zásadně liší.

Komunikátor byl po hardware stránce navržen s jistou mírou neurčitosti, co se týče použitých částí, které jsou reálně využity ve firmware. Jako první je možno zmínit nevyužívané tlačítka, jež v návaznosti na tuto skutečnost nebyly osazeny. V rámci komunikace s IQRF modulem, který je s MCU na desce plošných spojů propojen sběrnicemi SPI a UART, bylo pro komunikaci firmware zvolena komunikace po SPI, přičemž připojená sběrnice UART je na straně MCU ponechána bez využití s piny vykazujícími vysokou impedanci, a IQRF modul díky plug-inu pro komunikaci po SPI taktéž UART nevyužívá. Současně propojení digitálními signály IO a WAKE/BOND mezi IQRF a MCU je také nevyužito. Na straně GSM modulu bylo zapojeno pouze jediné komunikační rozhraní, kterým modul disponuje a tím je sériová linka. Ta je doplněna pomocnými signály pro hardware kontrolu přenosu - typickými pro RS232, které rovněž nejsou použity z důvodu přístupu FW k průběhu komunikace s GSM modulem, který je nevyžaduje. Pro GSM modul

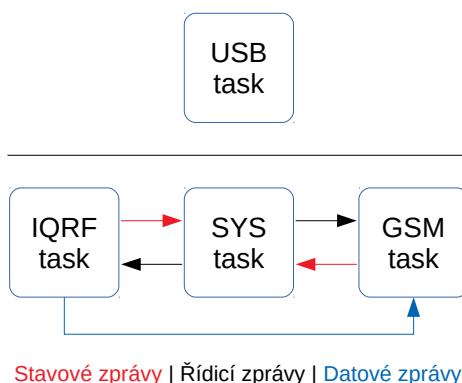
jsou dále propojeny dva digitální signály, kterými jsou VDD_EXT a STATUS, jejichž účelem je informovat MCU o stavu napájení GSM modulu. Dle datasheetu výrobce byl využit ve FW části pouze STATUS, jelikož disponuje vhodnějšími časovými výstupy pro toto použití.

Pro tvorbu a ladění firmware bylo využito vývojového prostředí Atollic TrueSTUDIO for ARM 8.1.0 a pro bezpečnost vývoje byl využit verzovací systém Git. Vytvořený pracovní projekt s firmware je přiložen v elektronické podobě k této práci. Autorem projektu s jeho základními funkcionalitami, jako je linker skript, soubory nutné pro zavedení firmware po resetu a hlavičkový soubor pro adresaci periférií konkrétního MCU, je Ing. Martin Stankuš, Ph.D, který tuto šablonu vytvořil v rámci svých aktivit pro VŠB-TUO. Tato šablona byla použita a dále rozvinuta ve výslednou aplikaci.

5.1 Struktura firmware

Navržený firmware komunikátoru pracuje na základě jádra operačního systému reálného času, kterým je FreeRTOS v10.2.0. Jelikož je úloha komunikátoru primárně založena na práci s daty, jsou hojně využity programové služby FreeRTOS nazývané fronty (Queues). Ty jsou využity také v některých vytvořených ovladačích periférií. Procesorové jádro zpracovává v provozním režimu čtyři úlohy (tasks), kterými jsou úloha IQRF, GSM, USB a systémová úloha. Dále byly vytvořeno sedm programových celků (modulů) využívajících některých funkcionalit RTOS. Tyto moduly jsou složeny vždy z dvojice zdrojových kódů - zdrojového a hlavičkového souboru. Moduly jsou: systémový, GSM, IQRF, USB, F-RAM, bateriový monitor a USB, které bylo pro svou náročnost jako jediné převzato z NXP SDK a přizpůsobeno architektuře FW. Jako součást práce byly vytvořeny vlastní ovladače pro následující periférie - GPIO, PORT, PCC, SCG, INTMUX, RTC, LPUART, LPSPI, LPI²C. Využita byla také periférie INTMUX, jejíž ovládání je zakomponováno v ovladačích ostatních periférií.

5.1.1 Inter-procesní komunikace



Obrázek 23: Vyobrazení IPC.

Úlohy podílející se na práci komunikátoru jsou naznačeny na obrázku 23, kde jsou vyobrazeny komunikační toky probíhající mezi úlohami (tasky). Ty můžeme nazývat také inter-procesními komunikacemi (IPC). Tato komunikace využívá vždy fronty pro předávání zpráv mezi úlohami, přičemž systémová úloha je jediná, která je zodpovědná za zasílání tzv. řídicích zpráv, kterými koordinuje průběh chování komunikátoru. Task IQRF a GSM naproti tomu posílají systémové úloze zprávy pouze stavové - informují tak o svém aktuálním stavu. Díky tomu je schopen systémový task reagovat například na vzniklou chybu. Výjimku v těchto komunikačních tocích tvoří datové zprávy, které jsou odesílány IQRF úlohou úloze GSM. Zde se jedná o přímé předávání naměřených dat k jejich odeslání směrem k datovému úložišti. USB task je jediný zcela oddělený od této IPC a to z toho důvodu, že je spuštěný pouze v servisním režimu, kdy je jedinou aktivní úlohou a stará se tak o veškerou funkčnost sám.

Systémový task je schopen reagovat na zprávy podle struktury IPC zpráv uvedené na výpisu 1, které jsou definovány v hlavičkovém souboru `app_system.h`, který je přidružený k systémovému tasku.

```
typedef enum
{
    APP_SYS_SENDER_GSM,
    APP_SYS_SENDER_IQRF,
} app_sys_sender_t;

typedef enum
{
    APP_SYS_STATE_NONE,
    APP_SYS_STATE_ERROR,
    APP_SYS_STATE_WORKING,
    APP_SYS_STATE_READY,
    APP_SYS_STATE_SLEEP,
    APP_SYS_STATE_RTC_INITED
} app_sys_state_t;

typedef struct
{
    app_sys_sender_t sender;
    app_sys_state_t state;
    uint32_t data;
} app_sys_ipc_t;
```

Výpis 1: Struktura komunikačních zpráv IPC přijímaných systémovým taskem.

IQRF task je schopen reagovat na zprávy podle struktury IPC zpráv uvedené na výpisu 2, které jsou definovány v hlavičkovém souboru `app_iqrf.h`, který je přidružený k IQRF tasku. Za povšimnutí stojí rozlišování příjemce těchto zpráv. Těmi může být dle datového typu `app_iqrf_receiver_t` buď samotný task nebo IQRF modul. Důvod pro toto rozlišení je v potřebách udržovat aktivní komunikaci s IQRF modulem v dobách, kdy mezi modulem a MCU neprobíhá užitečná komunikace. Tento požadavek plyne z datasheetu k IQRF modulu.

```
typedef enum
{
    APP_IQRF_CMD_NONE,
    APP_IQRF_CMD_INIT,
    APP_IQRF_CMD_SLEEP,
    APP_IQRF_CMD_WAKE,
    APP_IQRF_CMD_WAKE_RESET,
    APP_IQRF_CMD_PWR_ON,
    APP_IQRF_CMD_PWR_OFF,
    APP_IQRF_CMD_DEINIT,
    APP_IQRF_CMD_READ_INIT,
    APP_IQRF_CMD_LED_INDICATION_DISCOVERY,
    APP_IQRF_CMD_LED_INDICATION_MEASUREMENT,
    APP_IQRF_CMD_CHECK_NETWORK,
    APP_IQRF_CMD_COLLECT_DATA,
} app_iqrf_cmd_t;

typedef enum
{
    APP_IQRF_RECEIVER_TASK,
    APP_IQRF_RECEIVER_TR
} app_iqrf_receiver_t;

typedef struct
{
    app_iqrf_receiver_t receiver;
    app_iqrf_cmd_t command;
} app_iqrf_ipc_t;
```

Výpis 2: Struktura komunikačních zpráv IPC přijímaných IQRF taskem.

GSM task je schopen reagovat na zprávy podle struktury IPC zpráv uvedené na výpisu 3, které jsou definovány v hlavičkovém souboru `app_gsm.h`, který je přidružený ke GSM tasku.

```
typedef enum
```

```

{
    APP_GSM_CMD_NONE,
    APP_GSM_CMD_SLEEP,
    APP_GSM_CMD_WAKE_RESET,
    APP_GSM_CMD_WAKE,
    APP_GSM_CMD_PUBLISH_SERVICE_INFO,
    APP_GSM_CMD_PUBLISH_MEASURED_DATA
} app_gsm_cmd_t;

typedef struct
{
    app_gsm_cmd_t command;
    iqrf_comm_dpa_t dpa;
    uint32_t timestamp;
    uint8_t measured_data_size;
    uint8_t measured_data[APP_GSM_MEASURED_DATA_LEN];
} app_gsm_ipc_t;

```

Výpis 3: Struktura komunikačních zpráv IPC přijímaných GSM taskem.

Inter-procesní komunikace probíhá vždy přes veřejné funkce definované vždy ve zdrojovém kódu modulu, jež je příjemcem zprávy. Funkcemi jsou tedy:

- `app_sys_ipc_send(app_sys_ipc_t message)`
 - Umístěna v systémovém modulu.
- `app_iqrf_ipc_send(app_iqrf_ipc_t message)`
 - Umístěna v IQRF modulu.
- `app_gsm_ipc_send(app_gsm_ipc_t message)`
 - Umístěna v GSM modulu.

5.2 Využité periferie

Pro potřeby komunikátoru je využito především komunikačních periférií zajišťujících přenos dat. Tyto periferie však ke své správné funkci potřebují zdroje, jež jim jsou poskytovány pomocí jiných podpůrných periférií.

5.2.1 PORT

První z řady podpůrných přesto nutných periférií je PORT, která slouží pro konfiguraci konkrétních pinů MCU. Periferie slouží k:

- Povolení hodinového signálu
- Přiřazení obsluhující periferie
- Nastavení pull-up rezistorů

Periferie sama nevyžaduje inicializaci a je tedy využívána až v případě potřeby změny nastavení konkrétních pinů MCU. Funkcionalita je obsažena v ovladači GPIO.

5.2.2 SCG

Další podpůrnou periferií je SCG sloužící jako zdroj hodinových signálů pro celý MCU. Zde vytvořený ovladač periferie slouží především pro ověření nastavení a zjištění frekvence jednotlivých zdrojů hodinových signálů. Ovladač je využit napříč všemi ovladači pro komunikační linky, které potřebují pracovat se známou hodnotou frekvence hodinového signálu, jimiž jsou poháněny. Samotná periferie a tedy zdroje hodinových signálů jsou inicializovány ještě před spuštěním funkce main a to v obsluze přerušení na základě reset události. V práci jsou využity celkem tři zdroje hodinových signálů, kterými jsou:

- SIRC (Slow Internal Reference Clock)

$$- f_{SIRC} = 8 \text{ MHz}$$

- FIRC (Fast Internal Reference Clock)

$$- f_{FIRC} = 60 \text{ MHz}$$

- SOSC (System Crystal Oscillator)

$$- f_{SOSC} = 32\,768 \text{ Hz}$$

5.2.3 PCC

V návaznosti na periferii SCG slouží PCC periferie pro přiřazení konkrétního hodinového signálu z SCG do konkrétní periferie. Funkcionalita pro tuto periferii je zahrnuta v jednotlivých ovladačích periferií. Periferie neslouží pro nastavování parametrů hodinových signálů, ale pouze pro jejich propojení s periferiemi.

5.2.4 INTMUX

Kvůli velkému množství instancí periferií obsažených v KL28Z bylo zapotřebí sjednotit kanály pro přerušení těchto periferií. Tuto službu poskytuje periferie INTMUX, jež tak umožňuje až 32 přerušením přiřadit některé ze čtyř společných kanálů na výstupu periferie. Díky tomu není žádná z periferií ochuzena o schopnost budit jádro MCU. Naproti tomu mohou nevyužité instance periferií přenechat prostor jiným. Periferie INTMUX navíc nabízí možnost logických kombinací signálů přerušení z různých periferií za pomoci logických funkcí OR a AND.

5.2.5 GPIO

Periferie GPIO již nepatří mezi podpůrné, avšak svým charakterem se jedná o poměrně obecnou funkcionalitu. Periferie ovládá přidělené piny jako jednoduché digitální vstupy či výstupy. Periferie umožňuje následující funkcionality:

- Nastavení orientace pinu (vstup/výstup)
- Čtení digitální hodnoty na vstupu (0/1)
- Nastavení digitální hodnoty na výstup (0/1)
- Přepnutí logické hodnoty výstupu

GPIO je tak využito například pro práci s LED, DIP přepínačem, ale také jako řídicí vstup pro tranzistory ovládající napájecí obvody. Do ovladače této periferie je zkomponována funkcionalita pro periférii PORT a je využit napříč celým firmware. Samotná periferie, tak jako tomu bylo v případě PORT, nevyžaduje inicializaci a je využita až v případě potřeb jejích funkcionalit.

5.2.6 RTC

Periferie je zodpovědná za udržování hodnoty času, jak jej známe. Periferie pracuje v této práci s hodinovým signálem z externího krystalu o frekvenci 32 768 Hz a je tak schopna přesně čítat čas s periodou jedné sekundy. Periferie je využita pro tvorbu časových značek k jednotlivým měřením a zároveň slouží pro časování spouštění měření. Inicializace probíhá v systémovém tasku při prvním spuštění systému. Periferie udržuje hodnotu času i po resetu zařízení, avšak odpojení od zdroje energie tuto periférii resetuje a ztrácí se tak časová hodnota.

5.2.7 LPUART

Jedná se o periférii UART s ohledem na spotřebu energie tedy LP (low-power). Periferie umožňuje asynchronní komunikaci po sériové lince, přičemž umožňuje využívat signály RX, TX, CTS a RTS. V rámci této práce je využita pro komunikaci s GSM modulem s využitím signálů TX, RX a s rychlostí přenosu 9600 Bd. Pro periférii tvoří časovou základnu hodiny SIRC a konkrétní instance periferie LPUART2 komunikující s GSM modulem využívá pro svá přerušení periférii INTMUX. Periferie je inicializována za běhu na vyžádání.

5.2.8 I²C

Tak jako v případě LPUART i zde se jedná o periférii sloužící ke komunikaci pomocí sběrnice I²C v low-power provedení. Periferie je v práci využita pro komunikaci s ochranou prvního stupně akumulátoru neboli bateriového monitoru. Sběrnice využívá ke svému provozu dvou signálových linek. První z nich je linka hodinového signálu označována SCL, která je plně v režii master zařízení, druhou je linka datová označována SDA, která je v používání střídavě všemi

připojenými zařízeními na této sběrnici. V případě této práce je master sběrnice MCU a jediným slave připojeným zařízením je bateriový monitor. Periferie pracuje se základním hodinovým signálem ze zdroje SIRC a její hodinová linka je v rámci požadavků na komunikaci nastavena na frekvenci 100 kHz. Inicializace této periferie probíhá před startem systému.

5.2.9 LPSPI

Jedná se o poslední využívanou komunikační periférii taktéž v provedení low-power. Periferie využívá k provozu jednu linku pro hodinový signál, jednu linku pro datový přenos a X linek pro výběr zařízení, s nímž je požadována komunikace. V rámci této práce je periferie využita ve dvou instancích. První z instancí LPSPI komunikuje s F-RAM paměťovým modulem, druhá komunikuje s modulem IQRF. Vždy se jedná o komunikaci 1:1 a není tedy nutno přepínat mezi více zařízeními připojenými na jedné sběrnici. Obě instance využívají ke své funkci hodin SIRC, přičemž na signálové lince hodin je v případě instance komunikující s F-RAM frekvence 900 kHz, v případě IQRF 200 kHz. Průběh komunikace je pro obě instance zcela odlišný dle požadavků připojených modulů IQRF a F-RAM. Byly tak vytvořeny oddělené ovladače pro každou instanci LPSPI. V případě instance pro komunikaci s F-RAM probíhá její inicializace před startem systému, v případě druhé instance probíhá inicializace na požádání v průběhu funkce celého zařízení.

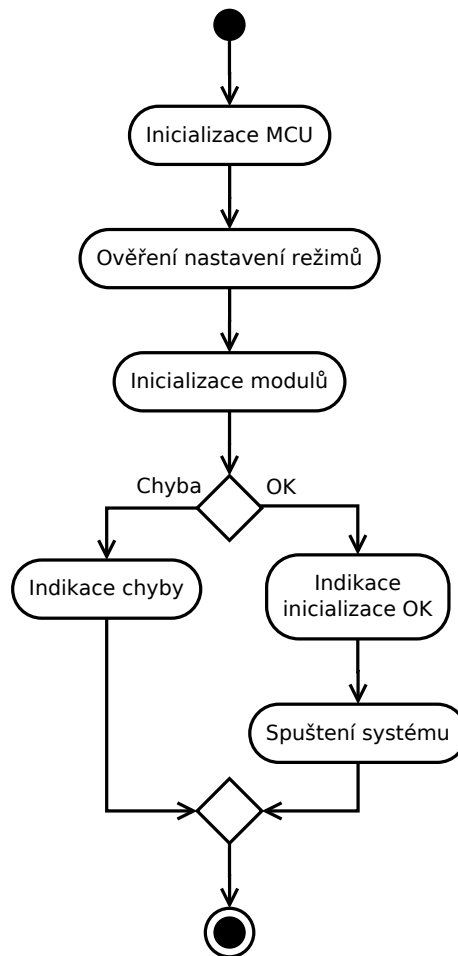
5.2.10 USB

Protokol USB je složen z několika úrovní, ty jsou poměrně rozsáhlé a tak bylo zvoleno převzetí ovladače této periferie z NXP SDK s malými úpravami pro kompatibilitu s komunikačními strukturami využitými v rámci tohoto firmware a RTOS. Periferie a její ovladač slouží pro zajištění veškeré potřebné komunikace s nadřazeným systémem a umožňují předávat datové zprávy oběma směry se zajištěním kompatibility protokolu USB. Použití je pak zaměřeno čistě na práci s užitečnými daty. Periferie využívá vstupního hodinového signálu FIRC a je inicializována před startem systému, avšak pouze v servisním režimu.

5.3 Základní chování komunikátoru

Firmware komunikátoru byl navržen pro činnost odpovídající diagramu 24. Diagram tedy popisuje aktivity vykonávané po resetu zařízení či po jeho úvodním spuštění. Jako první se provede inicializace nezbytně nutných prostředků MCU pro jeho funkčnost. Následuje ověření nastavení režimů - zde se jedná o jednorázové čtení DIP přepínače, jež je pro úsporu energie po přečtení odpojen od pull-up rezistorů a není dále využíván. Na základě zjištěného stavu se inicializují moduly pro konkrétní funkční režim potřebné. V poslední řadě proběhne ověření, zda se podařilo všechny součásti v pořádku nastavit a inicializovat načež se indikuje tento stav.

Indikace stavu komunikátoru je vizuální s pomocí čtyř LED zelené barvy nacházejících se u MCU. Po spuštění programu jsou všechny čtyři LED neaktivní dokud úspěšně neproběhne



Obrázek 24: Diagram aktivit chování komunikátoru.

inicializace všech modulů. Po úspěšné inicializaci se tedy všechny čtyři LED rozsvítí do dvou sekund od spuštění. Následuje spuštění systému a zde nutné systémové kroky probíhající pouze jednorázově po resetu, které jsou inicializační avšak již ne na úrovni HW ale na úrovni aplikační. V případě chyby v inicializační sekvenci dle diagramu 24 se systém nespouští a LED indikují, v tomto případě závažnou chybu nerozsvícením se, dokud není zařízení resetováno a spuštěno tak znovu. Pokud chyba přetrvává, nejedná se o běžný stav a je potřeba na zařízení vykonat servis na úrovni firmware.

Indikace chyby vzniklé za běhu představuje zhasnutí LED přidružené ke konkrétnímu funkčnímu modulu. Toto přidružení je následující:

- LED 8
 - Svítící indikuje IQRF modul v pořádku.
- LED 9
 - Svítící indikuje modul baterie v pořádku.

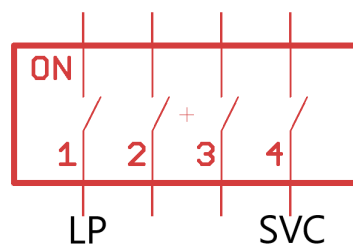
- LED 10
 - Svítící indikuje F-RAM modul v pořádku.
- LED 11
 - Svítící indikuje GSM modul v pořádku.

Reálné umístění LED na DPS napomáhá k rozpoznání přidružených modulů umístěním LED v porovnání s umístěním integrovaných obvodů ovládaných MCU. Tedy LED signalizující stav GSM firmware části je nejbližší umístěná LED ke GSM modulu na DPS. V případě, že moduly úspěšně provedou své inicializační průběhy, spouští se systém a LED indikují tento stav rozsvícením a následnou animací, která indikuje již aktivní spuštěný systém RTOS. Tato animace je postupné blikání všech LED v sekvenci. LED diody nejsou aktivní a tedy nesvítí v případě režimu nízkého odběru, kdy je komunikátor v režimu spánku. V případě, že probíhající animace neprobíhá souvislou sekvencí a některá z LED vychyluje optickou poslušnost, jedná se o indikaci chyby, která nastala za běhu. Takovou chybu se systém pokouší odstranit opětovnými pokusy o činnost, v níž chyba nastala.

5.4 Nastavení pracovního režimu

Pro volbu pracovního režimu je využit DIP přepínač. Je možno volit celkem tři pracovní režimy, kterými jsou - běžný režim, režim s nízkou spotřebou a servisní režim. Nastavení se provádí přepnutím daného bitu DIP přepínače do polohy ON. Běžný režim odpovídá nastavení přepínače na hodnotu "0000" tedy všechny bity v poloze OFF. Volba režimu nízké spotřeby je volena bitem 1 na hodnotu ON. Volba servisního režimu je volena bitem 4 do polohy ON. Toto je naznačeno na obrázku 25.

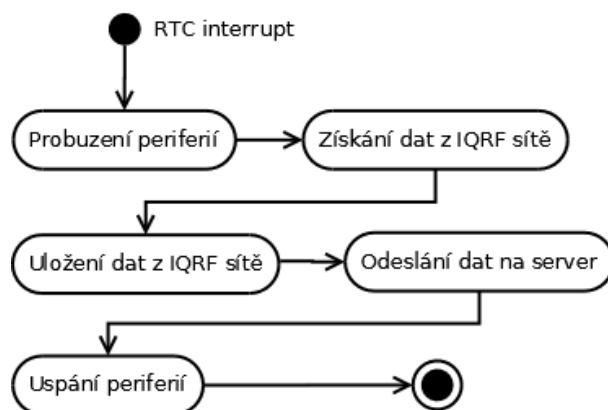
Nastavení pracovního režimu není během činnosti kontrolováno FW a je proto potřeba k docílení změny pracovního režimu zařízení resetovat a tím vynutit novou inicializaci. Reset je možno provést osazeným tlačítkem na DPS.



Obrázek 25: Nastavení režimů DIP přepínačem.

5.5 Průběh měřicího cyklu

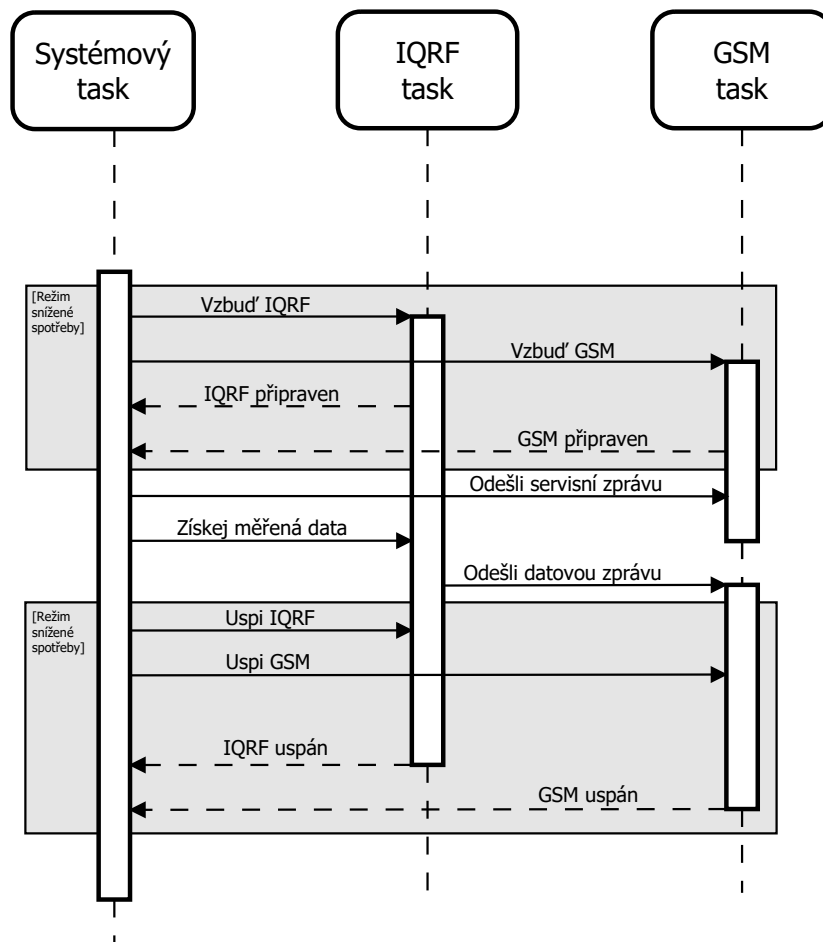
Měřicí cyklus je vyvolán vždy periodicky podle nastavení parametru v F-RAM paměti viz tabulka 3. Parametr měřicí periody je po nastartování systému vyčten z paměti a podle jeho hodnoty je provedeno nastavení periferie RTC, která se již stará o provedení indikace, že perioda vypršela a je tak potřeba provést měření.



Obrázek 26: Diagram aktivit měřicího cyklu.

Periferie RTC v přednastavených periodách vyvolá programové přerušení, které předává funkční semafor (FreeRTOS) jehož vyzvednutí systémovou úlohou umožní této úloze rozpoznat, že nastal čas provést měření. Úloha na základě tohoto, a v závislosti na zvoleném funkčním režimu - běžný či se sníženou spotřebou, provede předání příkazů pro ostatní úlohy, jež příkazy vykonají. Jedná se o příkazy probuzení IQRf/GSM modulů z režimu spánku, spuštění měření a spuštění přenosu dat směrem k serveru. Poté vyše systémová úloha příkazy k převedení do režimu spánku, pokud je vybrán režim se sníženou spotřebou.

Měřicí cyklus je vykonáván za kooperace všech vytvořených úloh, mimo úlohy USB, která je aktivní pouze v režimu servisním. Úlohy pro IQRf a GSM komunikace jsou řízeny systémovou úlohou, jež reaguje na vzniklé chyby a svou činností vytváří posloupnost měřicího cyklu, která je nastíněna diagramem 26. Typická bezchybná komunikace mezi úlohami během měřicího cyklu je naznačena diagramem 27, přičemž diagram zachycuje měřicí sekvenci typickou jak pro běžný režim provozu, tak pro režim provozu se sníženou spotřebou. Vyobrazené tmavé části diagramu jsou vykonávány pouze v případě aktivního režimu se sníženou spotřebou.



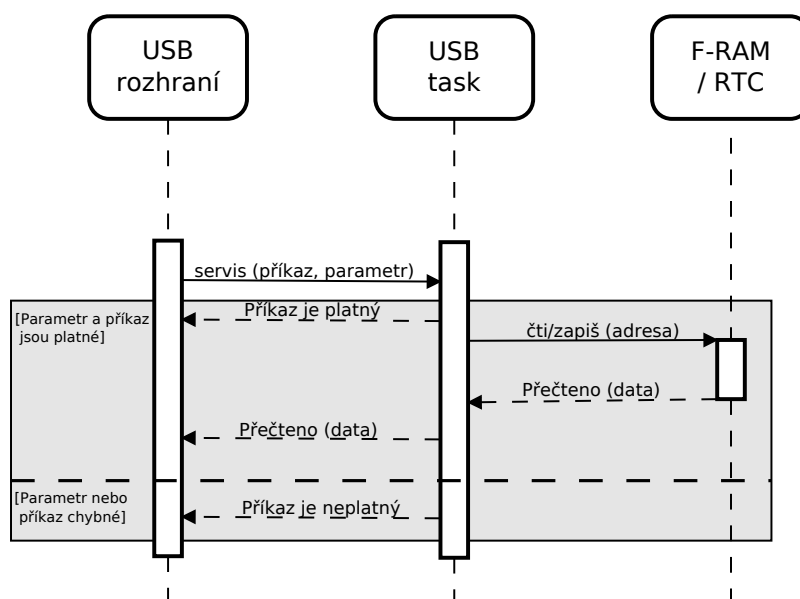
Obrázek 27: Komunikační diagram měřicích úloh.

5.6 Průběh servisního zásahu

V případě servisního režimu zařízení jsou standardní úlohy, aktivní v měřicích režimech, vypnuty a o funkcionalitu se stará pouze úloha USB. Ta je v kooperaci s USB ovladačem a zpracovává příchozí datové pakety. Uživatelský popis servisního režimu je uveden v kapitole 7. Firmware je schopen zpracovávat příkazy pro zápis či čtení servisních parametrů, přičemž vždy kontroluje pevně danou strukturu zprávy. Standardní parametry komunikátoru jsou uloženy v energeticky nezávislé paměti F-RAM a tudíž USB úloha s touto pamětí úzce spolupracuje. Zároveň servisní režim umožňuje práci s periferií RTC - pro nastavení či čtení času uchovávaného komunikátorem. Přístup k těmto periferiím je z vnějšího pohledu stejný. Průběh komunikace vyobrazuje diagram 28, kde zpráva s příkazem může obsahovat vždy jen jeden z příkazů - čtení/zápis. Na základě příkazu je proveden výpis dat v případě příkazu čtení, v případě příkazu zápisu nejsou zapsaná data již čtena ani navracena.

Tabulka 2: Mapa uspořádání F-RAM paměti.

Adresa	Délka dat (B)	Popis
0	16	ID zařízení
16	4	Perioda měření (sec)
32	2	Napětí akumulátoru (mV)
34	2	Kapacita akumulátoru (mAh)
36	2	Průměrný proud akumulátoru (mA)
38	2	Zbývající doba do vybití (min)
40	2	Stav nabití akumulátoru (%)
42	4	Datum měření akumulátoru (unix timestamp)
64	64	APN operátora
128	4	PIN kód
132	64	Adresa datového úložiště (server)
202	64	Cílový soubor POST HTTP požadavku (cesta ke skriptu)
266	64	Cíl serveru (host)
330	64	Content-Type parametr
394	1	Zadána je adresa datového úložiště ve formátu DNS (1) nebo IP adresa (0)
448	1	PNUM
449	1	PCMD
480	4	Adresa nejstarších měřených dat v paměti
484	4	Adresa nejstarších měřených dat v paměti
492 - 524232	70	Naměřená data (x 7481)

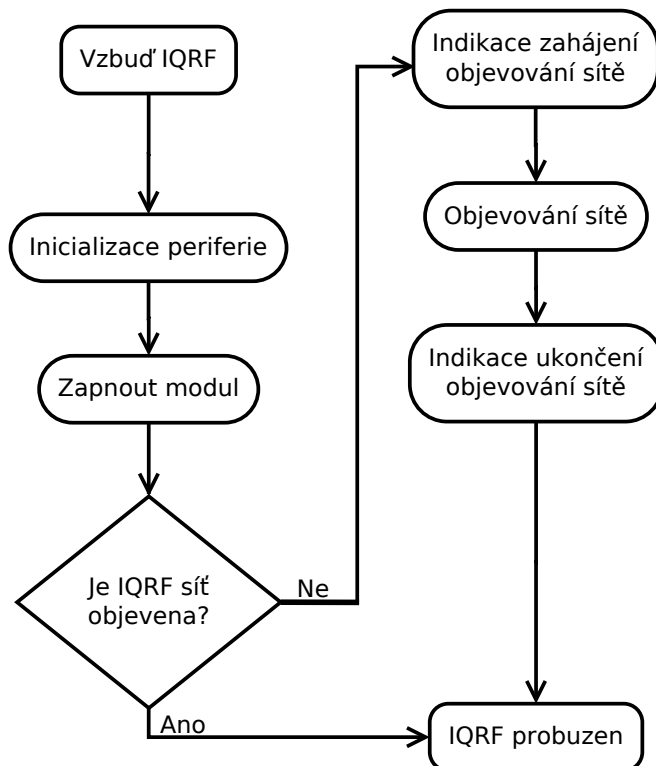


Obrázek 28: Komunikační diagram servisního režimu.

5.7 Průběh probuzení komunikátoru

Jak již bylo naznačeno diagramem 26, je při každém měřicím cyklu potřeba probudit jednotlivé komunikační komponenty a jim přidružené periferie, to za předpokladu, že je zařízení v provozním režimu se sníženou spotřebou. V režimu běžném nejsou úlohy uspávání a probouzení aplikovány.

Diagram 29 ukazuje posloupnost, která je vykonávána IQRf úlohou FreeRTOS po obdržení příkazové zprávy od servisní úlohy s příkazem probuzení. Task pracuje s IQRf modulem a periferií LPSPI. V diagramu se vyskytuje jedno větvení v případě znalosti IQRf mesh sítě. Toto dělení je založeno na skutečnosti, zda-li se jedná o první událost probuzení IQRf modulu - tedy událost po resetu zařízení, či se jedná o opakované probuzení v rámci měřicích cyklů. Task provádí tzv. discovery (objevování/výstavbu) sítě, což je mapování jednotlivých připojených uzlů sítě. Při opakování měření již není potřeba síť vždy objevovat znovu a proto je tato část vynechána. V případě výskytu jakékoli chyby během zpracovávání je provedeno opakování nezdařené akce ještě 3x před odesláním zprávy o chybě systémovému tasku.



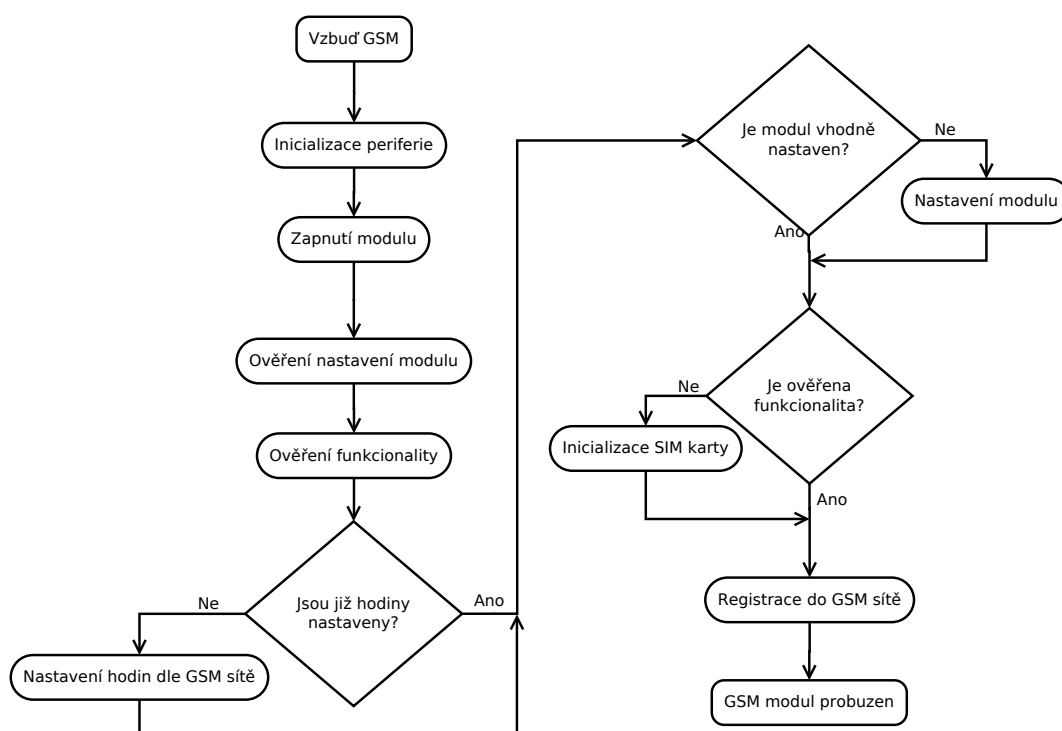
Obrázek 29: Diagram průběhu IQRF tasku vykonávající příkaz probuzení.

Na diagramu 30 je vyobrazená stejná posloupnost, avšak tuto vykonává GSM task a cílem je GSM modul a jemu přidružená periferie LPUART. Po inicializaci LPUART periferie je spuštěn modul. Samotné spuštění modulu je složeno z několika kroků:

1. Zapnutí napájení stabilizátoru na 4 V
2. Uvedení signálu PWR_KEY GSM modulu do logické 0
3. Vyčkání na logickou 1 u signálu STATUS GSM modulu
4. Uvedení signálu PWR_KEY zpět do logické 1

Signál PWR_KEY je výstupem GSM modulu a je jím standardně držen pull-up rezistorem v logické 1. Po spuštění modulu následuje fáze ověřování. V části ověřování nastavení modulu je od GSM modulu vyžádána jednoduchá odpověď a podle její struktury je vyhodnoceno, zda GSM modul již prošel obecným nastavením a má jej uloženo v energeticky nezávislé paměti pro další užití. Ověřování funkcionality představuje čekání po nastavenou maximální dobu čtyř sekund na asynchronně zasílanou zprávu ze strany GSM modulu s informacemi o tom, zda je modul provozu schopný bez dalších konfigurací. Jedná se zde také o ověření stavu SIM karty. Dalším krokem je nastavení hodin reálného času pro zařízení. To je inicializováno podmínkou, zda jsou již hodiny nastaveny. Tato podmínka je ověřena v periférii RTC, jelikož ta může být již

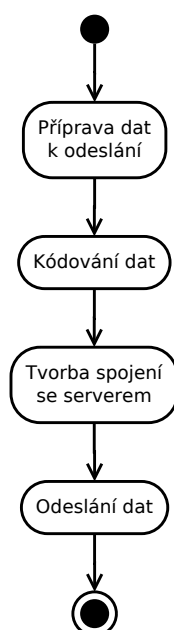
nastavena pomocí servisního zásahu. Pokud však toto neproběhlo a RTC nemá platně nastavené hodiny, potom právě GSM task v této fázi získává z GSM modulu čas sítě GSM a tento použije pro nastavení periferie RTC. V případě, že modul při úvodním ověřování neodpověděl na zprávu ve vhodném formátu, je provedeno obecné nastavení modulu. Totéž platí pro výsledek ověření funkcionality. Zde se pouze inicializuje SIM karta. Ověřuje se její vložení a případně se zadává PIN kód. Posledním krokem je registrace do GSM sítě, což završuje nastavení GSM modulu a zprovoznění jeho funkcionality pro následující použití. I zde platí, že v případě výskytu chyby během kteréhokoli z kroků je provedeno 3x opakování a následuje zaslání chybové zprávy systémovému tasku.



Obrázek 30: Diagram průběhu GSM tasku vykonávající příkaz probuzení.

5.8 Průběh odesílání dat datovému úložišti

Průběh odesílání dat GSM taskem je znázorněn diagramem 31 a je obecně platný jak pro zprávy datové tak i pro zprávy servisní odcházející na datové úložiště. V prvním kroku task sestaví zprávu (HTTP požadavek) podle pevně daného formátu, který je uveden na výpisu ???. Následně je celá datová zpráva kódována, aby neobsahovala GSM modulem zakázané znaky, které slouží jako řídicí při odesílání dat GSM modulem a tudíž, kdyby se nacházely v datové zprávě, by byly chybně vyhodnoceny a data nesprávně odeslána. Jakmile jsou takovéto data připravena k přenosu, je provedeno navázání spojení s datovým úložištěm, následované samotným přenosem dat. I zde platí, že jakýkoli výskyt chyby v průběhu sekvence je zachycen nejprve GSM taskem, který se pokusí 3x opakovat operaci a poté přechází k ohlášení chybového stavu systémovému tasku.



Obrázek 31: Diagram průběhu GSM tasku vykonávající příkaz odeslání dat.

5.8.1 Kódování zprávy před odesláním GSM modulem

Funkce pro kódování datové zprávy je jednou z více než 30 funkcí, které jsou využívány GSM taskem pro komunikaci s modulem využívajícím ovladače komunikační periferie. Funkce je založena na cyklickém procházení jednotlivých Byte zprávy a kontrole, zda se rovná některý z Byte právě zakázaným znakům. Zakázané znaky jsou uvedeny ve výpisu 5 spolu s kódovacím symbolem (APP_GSM_ENCRYPTOR). V případě, že je nalezena shoda, potom se provádí substituce tohoto Byte dvojicí Byte, je tedy potřeba datový prostor rozšířit, aby bylo možno zaměnit 1 Byte za 2 Byte. Zdrojový kód funkce je ve výpisu 4. V kódu je využita funkce `free_position_in_array()`,

kteřá má za úkol rozšířit pole tak, aby bylo možno na místo zakázaného Byte vložit 2 náhradní Byte.

```
static igg_return_t igg_gsm_forbidden_bytes_coding(drv_gsm_queue_element_t*
    data, uint32_t buffer_size, uint32_t* data_useful_size)
{
    igg_return_t ret = IGG_RETURN_DONE;

    for (uint32_t i = 0uL; ((i < *data_useful_size) && (IGG_RETURN_DONE == ret))
        ; i++)
    {
        if ((data[i] == APP_GSM_ASCII_ESC) || (data[i] == APP_GSM_ASCII_SUB))
        {
            // Found forbidden symbol.
            if (*data_useful_size < buffer_size)
            {
                free_position_in_array(data, buffer_size, i);
                data[i] = APP_GSM_ENCRYPTOR;
                data[++i] ^= APP_GSM_ENCRYPTOR;
                (*data_useful_size)++;
            }
            else
            {
                ret = IGG_RETURN_ERR;
            }
        }
    }
    return ret;
}
```

Výpis 4: Kódování zakázaných Byte.

```
#define APP_GSM_ASCII_ESC          (0x1B)
#define APP_GSM_ASCII_SUB          (0x1A)
#define APP_GSM_ENCRYPTOR           (0x55)
```

Výpis 5: Zakázané byte GSM modulu spolu s kódovacím znakem (encryptor).

6 Komunikace s datovým úložištěm

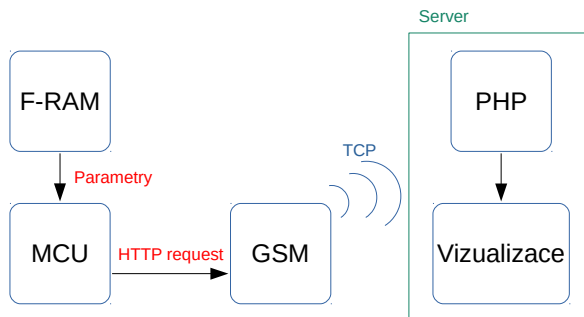
Jako datové úložiště pro tuto práci, jak již bylo naznačeno v kapitole 3.2, byl zvolen webový server. Pro přenos dat je využit HTTP POST požadavek ve specifikovaném tvaru. Pro příjem informací na straně serveru byl vytvořen skript psaný v jazyce PHP, který má za úkol zpracovávat předdefinovaný formát příchozích HTTP požadavků, tato zpracovaná data následně vizualizovat. HTTP požadavek odesílaný zařízením má formát dle výpisu ???. Označením <LF> je znázorněn ASCII znak zalomení řádku tedy 0x0A. Ve výpisu kódu ??? jsou červeně znázorněny části, které jsou uživatelsky definovatelné a je možno je měnit a přizpůsobit tak komunikaci s datovým úložištěm podle potřeb. Zbylé části jsou definovány firmwarem a není možno provést jejich změnu bez zásahu do kódu.

```
POST _skript_ HTTP/1.1<LF>
Host: _target_<LF>
Content-Type: _content_type_<LF>
Content-Length: _length_<LF>
<LF>
_data_<LF>
```

Výpis 6: HTTP request

Část HTTP požadavku ??? označená jako `_skript_` slouží pro uvedení cesty ke skriptu zpracovávajícímu požadavek. Část `_target_` slouží pro specifikaci serverového cíle (název domény) pro virtuální hostování případně také port, část `_content_type_` určuje obsah HTTP požadavku, část `_length_` je vypočítána firmwarem a představuje počet byte sekce `_data_`, která již obsahuje samotný užitečný obsah zprávy. Číselná hodnota délky je zde transformována do série znaků představující tuto hodnotu.

Celý tento textový řetězec je sestaven MCU podle šablony uvedené ve výpisu ??? a to na základně nastavení uživatelských parametrů, které MCU získá čtením z energeticky nezávislé paměti. Vše je spolu s řídicími příkazy předáno GSM modulu, který se stará o přenos těchto dat za pomoci protokolu TCP.



Obrázek 32: Schématické znázornění komunikace s datovým úložištěm.

Pro zpracování obsahu POST požadavku byl vytvořen skript, který provádí rozlišení typu zprávy (servisní/datová) a na základě vytvořeného protokolu těchto typů zpráv je schopen rozeznat přijímaná data a ty patřičně přeložit a reprodukovat. Skript provádí interpretaci přijatých zpráv do textové podoby ukládáním do specifikovaného souboru, kterým může být webová stránka či čistě textový soubor. V případě potřeby využít databázi je možno velice snadno upravit vytvořený skript pro zápis do tabulek díky již hotovému překladu dat do patřičných datových typů.

Přenášená data jsou v binární podobě, která je datově nenáročná a jsou díky tomu data kratší, což je především znatelné u číselných hodnot. Díky tomu mohou být i velká čísla přenášena například pouze čtyřmi Byte, čímž je ušetřena datová komunikace ale také doba odesílání dat, což v dlouhodobém časovém měřítku může mít vliv na spotřebu zařízení. Je tedy potřeba správně interpretovat jednotlivé skupiny Byte tak, aby výstup odpovídal přenášeným datům. Skript poskytuje tuto funkcionalitu, která je nutná ke správnému čtení dat, jelikož při binárním přenosu nejsou data čitelná.

6.1 Protokol servisních a datových zpráv

Popis protokolu servisních zpráv dle výpisu 7 je uveden níže. Symboly `_` uvedené ve výpisu nemají funkční význam a slouží pouze pro grafické oddělení jednotlivých částí. V reálné zprávě se mezi jednotlivými částmi nenachází nic a zpráva je souvislý řetězec Byte.

```
G=_s_device_voltage_current_capacity_time_soc_date  
G=_d_device_nadr_pnum_pcmd_date_data
```

Výpis 7: Protokol servisní zprávy.

- G=
 - Prefix/klíč uvozující celou zprávu.
- s/d
 - Znak určující obsah zprávy. Kde ‘s’ představuje servisní zprávu a ‘d’ zprávu datovou.
 - 1 B
- device
 - Textový řetězec označující zařízení, jež zprávu odeslalo.
 - 16 B
- voltage
 - Hodnota napětí akumulátoru v mV.

- 2 B
- current
 - Průměrná hodnota proudu z/do akumulátoru v mA. Záporná hodnota zde indikuje vybití akumulátorů, kladná nabíjení.
 - 2 B
- capacity
 - Zbývajících kapacita akumulátoru v mAh.
 - 2 B
- time
 - Odhadovaná doba do vybití akumulátoru v minutách.
 - 2 B
- soc
 - Stupeň nabití akumulátoru v %.
 - 2 B
- date
 - UTC datum a čas odpovídající době měření výše uvedených parametrů ve formátu Unix Timestamp.
 - 4 B
- nadr
 - Adresa senzoru IQRF sítě, které odpovídají data ve zprávě.
 - 1 B
- pnum
 - Označení periferie senzoru IQRF sítě.
 - 1 B
- pcmd
 - Příkaz využitý k získání dat z periferie senzoru IQRF sítě.
 - 1 B
- data
 - Konkrétní data proměnné délky odpovídající datům senzoru IQRF sítě.
 - x B

6.2 Zpracování zpráv PHP skriptem

Zpracování staví na základě známých umístění jednotlivých částí zprávy a známé délky každé části. Je tedy předem známo, jaká část zprávy představuje jakou funkční část. Skript pomocí těchto definovaných parametrů vyjme daný počet Byte a utvoří z něj reprezentovanou hodnotu. Funkcionalita skriptu také umí dešifrovat příslušné Byte, které jsou díky GSM modulu zakázány a nemohou být odeslány.

Zdrojový kód skriptu je přiložen v elektronické kopii této práce.

Received service message at server date: 04-04-2019 18:57:59 Sender: IGG_v1.10a Voltage = 8113 mV Average current = -101 mA Remaining capacity = 6172 mAh Remaining time to battery empty = 3666 min Battery state (SOC) = 100 % Date of measurement: 04-04-2019 18:57:52 UTC	Received measurement message at server date: 04-04-2019 18:59:10 Sender: IGG_v1.10a NADR = 0xe7 PNUM = 0x0a PCMD = 0x80 Measured data = 0x17 0x70 0x01 Date of measurement: 04-04-2019 18:58:34 UTC
--	---

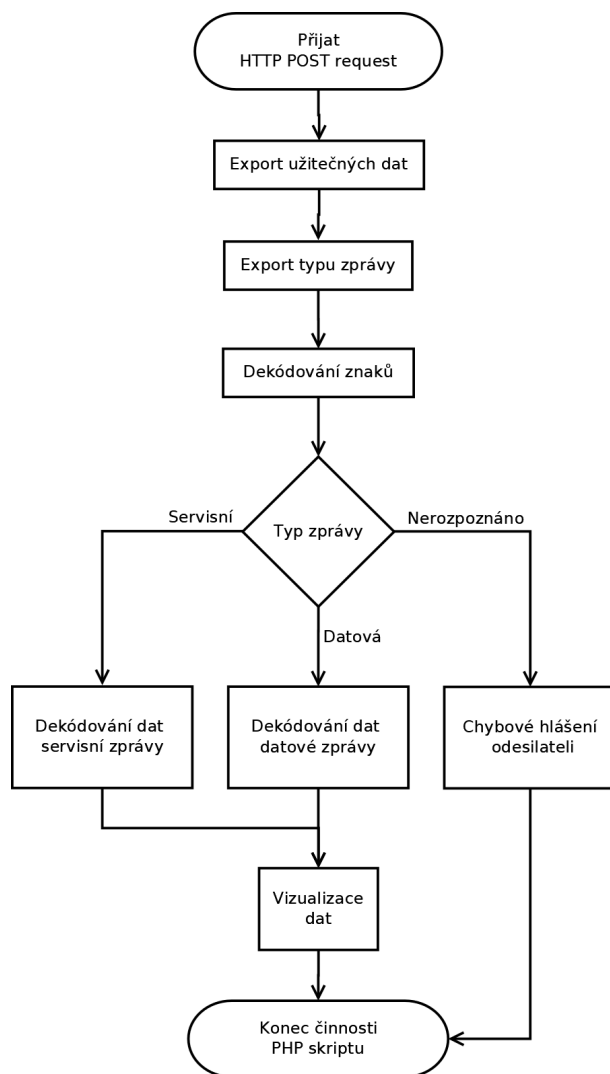
Obrázek 33: Ukázka výstupu zpracovaných dat na straně serveru (vlevo servisní zpráva, vpravo datová).

Skript nejprve provede vyjmutí užitečných dat z požadavku na základě známého klíče. Následuje získání informace o typu zprávy (servisní/datová) a dekodování znaků, jež byly zakódovány komunikátorem kvůli nemožnosti jejich použití v rámci GSM modulu. V rámci znalosti typu zprávy se provede extrakce jednotlivých částí zprávy a jejich převod do správných datových typů. Data jsou poté zapsána do souboru.

Cílový soubor, do něhož skript v části vizualizace data zapisuje, je možno změnit. Je možné také konfigurovat parametry zápisu do souboru. V tomto případě je využito přidávání nově příchozích dat na konec souboru. Cílový soubor tak obsahuje data dodržující časovou posloupnost podle příchozích zpráv. Úpravu těchto parametrů je možné provést v definicích vypsáných výpisem 8.

```
define('FILENAME', 'index.html');  
define('PUT_PARAMS', FILE_APPEND | LOCK_EX);
```

Výpis 8: Parametry PHP skriptu pro zápis do souboru.



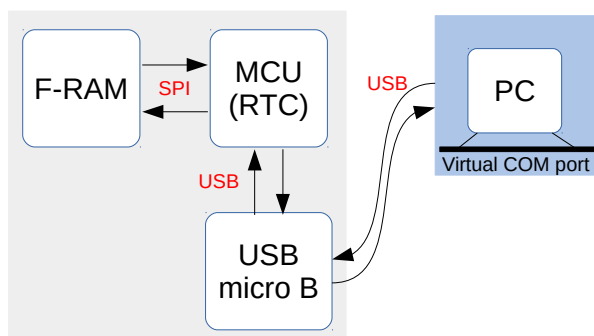
Obrázek 34: Vývojový diagram zpracování HTTP PHP požadavku PHP skriptem.

7 Servisní konfigurace

7.1 Servisní nastavení komunikátoru

Jelikož se nejedná o primitivní zařízení, je potřeba aplikovat chování podle požadavků uživatele. Toto uživatelské nastavení musí přetrvávat až do jeho další změny a proto je zapotřebí jej uchovávat v energeticky nezávislé paměti, tedy v případě této práce v paměti F-RAM. Pro tyto účely byl vytvořen jednoduchý komunikační protokol, který je přenášén mezi uživatelským PC a komunikátorem za pomoci osazeného micro USB konektoru. Je tedy zapotřebí USB A - USB B micro kabelu pro připojení. Komunikátor se standardně ohlásí díky USB periférii integrované v MCU uživatelskému PC jako běžný COM port. Následně je možno za použití libovolného software komunikovat s komunikátorem pomocí vytvořeného protokolu. V nejjednodušším případě postačí využít na PC libovolný terminálový software.

Výjimku v případě úložiště parametrů tvoří hodnota času komunikátoru. Jelikož je čas proměnný, není možné jej ukládat do paměti a proto je ukládán či čten z periférie RTC, jež se stará o jeho aktuálnost po celou dobu připojeného zdroje napájení. Při výpadku či odpojení napájení je tato hodnota ztracena. Avšak pokud není nastavena servisním zásahem, komunikátor je schopen ji získat ze sítě GSM.



Obrázek 35: Schématické znázornění servisního zapojení.

Samotný protokol se skládá ze dvou typů komunikačních zpráv - zprávy pro čtení a zprávy pro zápis. Tyto zprávy jsou přeneseny přes USB protokol do MCU, který je interpretuje a vykoná patřičný úkon. Vykonávané úkony odpovídají typům zprávy a těmi jsou čtení či zápis z/do F-RAM paměti, která obsahuje veškeré nastavitelné parametry komunikátoru mimo čas. Označením <LF> a <CR> jsou znázorněny ASCII znaky odpovídající hodnotám 0x0A a 0x0D pro <CR>.

```
IGGR<CR><LF>_item_<CR><LF>
IGGW<CR><LF>_item_<CR><LF>_data_<CR><LF>
```

Výpis 9: Protokol servisních příkazů pro konfiguraci komunikátoru.

Popis protokolu servisních zpráv dle výpisu 9 je uveden níže. Zpráva je složena z tisknutelných znaků pro usnadnění při práci s terminálovými programy což znamená, že i číselné hodnoty je možno zapisovat jako sadu znaků, přičemž až MCU tyto znaky rozpozná a přeloží na odpovídající číselnou hodnotu do číselného datového typu. Totéž funguje v opačném směru - tedy MCU číselné hodnoty přeloží na sadu znaků reprezentující hodnotu v čitelnější podobě.

- IGG
 - Prefix uvozující celou zprávu.
- R/W
 - Znak určující typ zprávy.
 - R - zpráva požaduje čtení dat.
 - W - zpráva požaduje zápis dat.
- __item__
 - Cílová položka.
 - Složeno vždy ze 2 číselných znaků.
 - Položky uvedeny v tabulce 3.
- __data__
 - Data pro zapsání.
 - Tato položka je nutná u příkazů pro zápis. U příkazů pro čtení je standardně ignorována.
 - Délka dat je v MCU ověřena.

Tabulka 3: Položky pro servisní konfiguraci komunikátoru.

Položka	Význam	Typ
00	Název zařízení	string (16 B)
01	Perioda měření (sec)	unsigned int (4 B)
02	Napětí akumulátoru (mV)	unsigned short (2 B)
03	Zbývající kapacita akumulátoru (mAh)	unsigned short (2 B)
04	Průměrný proud akumulátoru (mA)	short (2 B)
05	Zbývající čas do vybití akumulátoru (min)	unsigned short (2 B)
06	Stav nabití akumulátoru (SOC) (%)	unsigned short (2 B)
07	Timestamp měření parametrů akumulátoru	unsigned int (4 B)
08	GSM APN	string (64 B)
09	SIM PIN	string (4 B)
10	Adresa serveru (IP/DNS)	string (64 B)
11	Cíl POST požadavku	string (64 B)
12	Host	string (64 B)
13	Content-Type	string (64 B)
14	Adresa serveru ve formě DNS(1) nebo IP(0)	byte (1 B)
15	PNUM	byte (1 B)
16	PCMD	byte (1 B)
17	Adresa nejstarších měřených dat v F-RAM	unsigned int (4 B)
18	Adresa nejnovějších měřených dat v F-RAM	unsigned int (4 B)
99	RTC (Interní hodiny komunikátoru)	unsigned int (4 B)

Po odeslání zprávy definované dle výpisu 9 MCU jako první odešle odpověď do uživatelského PC o potvrzení, že zprávu rozpoznal a je korektní. Tato odpověď je znázorněna na výpisu 10 pro oba případy - správně rozpoznanou zprávu a nerozpoznanou zprávu. V případě, že se jedná o zprávu typu čtení dat následují tuto odpověď data aktuálně uložená v paměťovém prostoru.

Command OK <CR><LF>

Command ERROR <CR><LF>

Výpis 10: Odpověď na přijatou servisní zprávu.

Díky tomuto protokolu postačuje využít libovolného datového terminálu a standardního USB kabelu pro přizpůsobení komunikátoru. Nastavit je možné položky pro GSM modul - tedy parametry připojení k serveru, položky pro IQRF síť - dotazovanou periferii a příkaz pro tuto periferii, položky samotného zařízení a je možné také číst uložená data v paměti zařízení. Příklad průběhu servisní komunikace je vyobrazen ve výpisu 11, který obsahuje nastavení nového názvu zařízení a jeho následného ověření vyčtením. Červeně jsou ve výpisu vyznačeny odpovědi komunikátoru směrem k uživatelskému PC.

```
IGG_W<CR><LF>00<CR><LF>device name A<CR><LF>
```

```
Command OK <CR><LF>
```

```
IGGR<CR><LF>00<CR><LF>
```

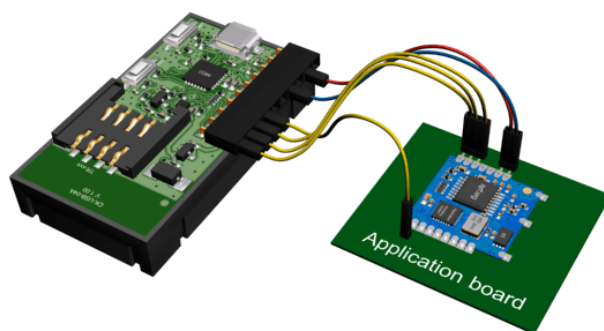
```
Command OK <CR><LF>
```

```
device name A<CR><LF>
```

Výpis 11: Příklad servisní komunikace.

7.1.1 Servis IQRF mesh sítě

Samotná IQRF síť je také konfigurovatelná a v rámci komunikátoru je to také umožněno. Servisní nastavení IQRF sítě probíhá skrze centrální uzel sítě, kterým je koordinátor, v tomto případě vytvořený komunikátor, respektive jeho IQRF modul. Servisní zásahy jsou velice uživatelsky přívětivě situovány ve vývojovém software IQRF IDE. V rámci využití tohoto software byl vyveden servisní konektor IQRF modulu výlučně určený právě pro konfiguraci IQRF MESH sítě. Jedná se o 6pinový konektor na DPS označený jako IQRF PROGRAMMING. Pro konfiguraci je zapotřebí nejprve přivést komunikátor do reset stavu za pomoci DIP přepínače MCU RESET, nebo odpojením akumulátoru. Dále je potřeba přemostit pomocí jumperu konektor napájení IQRF modulu J5 a propojit tak piny 2 a 3 namísto standardního provozního propojení pinů 1 a 2. Tím je zajištěno napájení modulu z externího programátoru. Ke konektoru je pak potřeba pomocí 6žilového kabelu připojit IQRF programátor CK-USB-04A a ten propojit s uživatelským PC. Poté je možno provádět běžnou konfiguraci pomocí IQRF IDE. Zde se jedná o konfiguraci, která není v rámci konfigurace komunikátoru možná, a jedná se čistě o konfiguraci IQRF sítě. Jde především pro vytváření IQRF sítě - přidávání a odebírání uzlů do sítě, které jsou běžným zásahem například při rozšíření měřicí sítě, což servisní komunikace komunikátoru dle 7.1 neumožňuje.



Obrázek 36: Naznačení servisní konfigurace IQRF MESH sítě. [15]

8 Spotřeba elektrické energie

Spotřeba elektrické energie zařízení je závislá na spuštěném pracovním režimu zařízení. Jelikož servisní režim není standardní a jeho využití je předpokládáno jako výjimečné, není spotřeba v tomto režimu podstatná. Přesto je předpoklad spotřeby v tomto režimu nižší než spotřeba v normálním pracovním režimu a to především proto, že režim využívá k provozu pouze MCU a F-RAM paměť a tedy IQRF a GSM moduly nejsou vůbec napájeny.

Obecně lze označit GSM modul za nejméně energeticky úspornou jednotku komunikátoru. Tato energetická náročnost se projevuje při komunikaci s GSM sítí, kdy nastávají špičkové odběry energie. Díky zapojení ochranného obvodu prvního stupně pro akumulátor, který umožňuje sledování stavu akumulátoru, bylo využito dat tímto obvodem naměřených k sestavení typické charakteristiky spotřeby energie v normálním režimu a v režimu snížené spotřeby komunikátoru.

Spotřeba energie v obou režimech byla měřena v náročném provozním režimu co se týče nastavení měřicí periody. Komunikátor vykonával sběr dat z IQRF sítě s periodou jedné minuty, přičemž s touto periodou data vždy také odeslal na datové úložiště. Naměřené hodnoty proudového odběru byly odečítány ze servisních zpráv komunikátorem odeslaných. Obvod uskutečňoval měření parametrů akumulátoru s periodou 250 ms, přičemž hodnota odečtená ze servisní zprávy představuje průměrnou hodnotu z periodické doby odeslání dat na datové úložiště - tedy průměr za 1 min.

Tabulka 4: Spotřeba elektrické energie.

Režim	Maximální odběr (mA)	Minimální odběr (mA)	Odhadovaná doba provozu (h)
Normální	134	98	(47 - 65)
Snížená spotřeba	60	45	(106 - 142)

Hodnoty z tabulky 4 ukazují, že režim nízké spotřeby výrazně prodlužuje provozuschopnost komunikátoru a to o dvojnásobek času. Zároveň jsou tyto údaje teoreticky horším případem provozu komunikátoru, jelikož se předpokládá využití s měřicími intervaly delšími než 1 min jak je tomu u těchto naměřených hodnot. Tabulka uvádí odhadovanou dobu provozu zařízení - tyto hodnoty jsou teoretickým výpočtem naměřené spotřeby proudu s předpokladem, že kapacita akumulátoru je 6400 mAh, což je skutečná nominální kapacita akumulátoru tohoto zařízení.

9 Obsluha zařízení

9.1 Příprava zařízení pro start

Pro uvedení do provozu je potřeba jako primární zdroj energie připojit akumulátor a to pomocí konektoru se zámkem J7. Dále je pro správnou funkčnost nutné mít vloženou SIM kartu a to buď bez nastaveného kódu PIN, nebo s nastaveným kódem spolu se správnou konfigurací (zadaným PIN kódem v zařízení pomocí servisního nastavení). Pro správnou funkčnost je nutné mít připojenou anténu pro GSM modul.

9.2 Změna pracovního režimu

Pro uvedení zařízení do jednoho z pracovních režimů (běžný režim, režim snížené spotřeby, servisní režim) je potřeba zajistit správné nastavení čtyř bitového DIP přepínače označeného na DPS jako MCU CONFIG. Změna nastavení provozního režimu vyžaduje reset komunikátoru. To není vyžadováno v případě, že je toto nastavení aplikováno před spuštěním zařízení dle kapitoly 9.3. Nastavením správných bitů DIP přepínače je zvolen provozní režim a kapitola 5.4 uvádí režimy pro jednotlivé konfigurace.

9.3 Uvedení zařízení do provozu

Start zařízení se provádí stiskem a přidržením tlačítka POWER ON umístěného na druhé straně desky plošných spojů oproti konektoru pro připojení akumulátoru J7. Tlačítko je potřeba držet do doby než se rozsvítí čtyři zelené LED umístěné u MCU. Toto rozsvícení indikuje provoz zařízení.

Aby bylo možné zařízení správně provozovat je zapotřebí před jeho spuštěním provést konfiguraci dle kapitoly 7 v servisním provozním režimu zařízení.

Zařízení pro svou bezchybnou činnost potřebuje, aby před jeho spuštěním bylo připojeno do IQRf sítě s aktivními uzly. Po startu zařízení provádí první inicializační měření a v případě nedostupnosti žádného z uzlů v síti IQRf zařízení hlásí chybu.

9.4 Reset zařízení

V některých případech je zapotřebí zařízení resetovat. Jsou jimi změna pracovního režimu, vynucení opětovné inicializace zařízení nebo konfigurace IQRf modulu skrze programovací IQRf konektor s využitím IQRf IDE. Reset je možno okamžitě aplikovat tlačítkem MCU RESET na DPS. Toto tlačítko je určeno k momentovému resetu pro výše uvedené případy použití. Paralelně k tlačítku je možno také využít DIP přepínač se stejným účinkem. Tento je určen pro možnost udržet zařízení v reset stavu po dobu, kdy je přepínač v poloze ON. Tento přepínač je vhodné využít během programování IQRf modulu pomocí externího programátoru, přičemž není nutné v případě, že akumulátor není připojen.

9.5 Vypnutí zařízení

Pro vytvořený komunikátor neexistuje možnost jeho vypnutí. Pokud je potřeba ukončit jeho činnost standardně postačuje odpojit akumulátor. Toto nejlépe provádět v době, kdy nejsou moduly aktivní - zařízení při aktivitě indikuje svou činnost pomocí LED, přičemž každý z modulů má vlastní LED, které je vhodné před odpojením zkontrolovat.

10 Zhodnocení výsledků a závěr

Práci se podařilo dokončit v plném rozsahu zadání, bylo navrženo elektrické zapojení vybraných komponent s ohledem na spotřebu elektrické energie, byl vytvořen funkční firmware s ohledem na spotřebu elektrické energie, komunikační protokol pro komunikaci mezi zařízením a datovým úložištěm, dále mezi servisním nadřazeným systémem a komunikátorem. Fyzicky tak vzniklo elektronické zařízení s funkcí komunikátoru, obhospodařující měření minimálně jednoho uzlu v síti IQRf, a reprodukující naměřená data směrem k datovému úložišti s využitím datových sítí GSM. Zařízení využívá akumulátoru jako zdroje energie a je tak schopno operovat i mimo dosah stálých zdrojů elektrické energie. Na straně datového úložiště byl vytvořen skript odpovídající komunikačnímu protokolu, jež interpretuje datové a servisní zprávy komunikátoru do dále prezentovatelné podoby.

V rámci návrhu a vývoje tohoto zařízení vznikly celkem tři verze hardware, přičemž pouze verze 1.2 a 1.3 byly vyrobeny. Práce popisuje hardware pouze verze 1.3, jelikož jde o nejvyšší variantu. Cesta návrhu a tvorby hardware pro komunikátor byla poměrně složitá, díky nezkušenosti a několika nejasnostem v dokumentacích výrobců použitých součástek vzniklo několik chyb, které vedly k výrobě dvou verzí hardware. První vyrobená verze 1.2 obsahovala složitější propojení komunikačních linek modulů GSM a bateriového monitoru. Tyto totiž byly doplněny o obvody pro komunikace datových linek s jinými napěťovými úrovněmi, které byly v poslední verzi vypuštěny. Toto usnadnění vedlo také k odstranění napěťového regulátoru na 1,8 V, který sloužil jako reference pro obvody na úpravu napěťových úrovní. Dále se jednalo například o přidání zpětné vazby z GSM modulu do MCU pro možnost zjištění stavu zapnutí GSM modulu, přidání startovacího tlačítka pro celé zařízení a další opravy chyb vzniklé tvorbou vlastních knihoven použitých součástek.

Během realizace hardware také vznikaly překážky jako například špatně dodané součástky, nebo nefunkční kusy součástek, poškození součástek tepelným vlivem pájení, studené spoje po pájení nebo naopak címem spojené plošky integrovaných obvodů. Samotné osazování a pájení, probíhající svépomocí spolu s následnou diagnostikou vzniklých vad v laboratoři VŠB FEI bylo rovněž časově náročné. Všechny tyto záležitosti vedly ve výsledku k časové náročnosti dokončení komunikátoru v řádném termínu.

Na straně firmware vzniklo také několik verzí a ty se měnily se vznikem nových potřeb v používání modulů a periférií. Až výsledná verze obsahuje RTOS a má veškeré potřebné komponenty plně provozuschopné. Samotná tvorba firmware však již nebyla z pohledu chyb kritická. Nejdéle trvající činností ve vývoji firmware byla tvorba ovladačů periférií MCU. Ty musejí splňovat nároky kladené externími moduly spolu se správným přístupem k používání periférií MCU, které díky svým rozšířeným schopnostem dělají ovladače komplikovanější. Možnost využít SDK výrobce by rapidně zkrátilo čas vývoje, avšak to nebylo využito kvůli vadám zmíněným v kapitole 5. Dalším poměrně náročným úkolem byla komunikace s GSM modulem. Ta probíhá pomocí AT příkazů, které jsou poměrně rozšířené pro tyto typy modulů. Zachytávání asynchronně přichozích

zpráv v případech, kdy se nejednalo o odpovědi na požadavky MCU, ale o přenosy inicializované samotným GSM modulem komplikovalo komunikaci a to především v případech, kdy GSM modul asynchronně zasílal zprávy během probíhající komunikace jiného účelu. Samotná komunikace pomocí AT příkazů obsahuje velké množství znakových řetězců, které musejí být během komunikace rozeznávány a naopak také stavěny dle formátů odpovídajících datasheetu, a které tak navýšily objem kódu a také náročnost na paměť. Aplikační část firmware zabývající se komunikací s GSM modulem včetně GSM tasku tak svým objemem dosahuje téměř 1500 řádků zdrojového kódu bez hlavičkových souborů, obsahujících definice AT příkazů.

Komunikátor vykazuje dle kalibrovaného integrovaného bateriového monitoru hodnoty spotřebovaného proudu, které jsou velice přívětivé. Značná úspora energie nastává také použitím pracovního režimu se sníženou spotřebou, který se předpokládá jako nejpoužívanější. Výsledná spotřeba zařízení je však úzce spjata s provozním nastavením. Konkrétně se zde jedná o počet připojených uzlů IQRF MESH sítě a o periodu měření. Množství připojených měřicích zařízení spotřebu energie navyšuje díky delším komunikačním časům, při nichž nastávají největší odběrové špičky.

V aktuální verzi hardware i firmware je stále prostor pro změny a vývoji k dosažení vyšší kvality zařízení. Náměty na tyto možné budoucí změny jsou uvedeny níže.

Hardware část obsahuje následující nedostatky: programovací konektor modulu IQRF J4 má zaměněné pořadí signálů SDI a SDO, neosazený konektor F-RAM J13 má chybně vytištěné popisy signálů, konektoru pro volbu zdroje napájení IQRF modulu J5 chybí popisy, umístění konektorů pro antény J2, J10 by bylo vhodnější blíže k okraji DPS s ohledem na vyrobiteľnost, DPS by mohla být širší pro zmenšení vůle mezi krabičkou a akumulátorem, konektory pro ladění komunikačních sběrnic (J9, J13, J14) nejsou u dalších kusů potřeba. Dále by bylo vhodné zvážit a otestovat schopnost GSM modulu pracovat na napájecím napětí 3,3 V pro zjednodušení napájecích cest a snížení množství komponent spolu s možností využít vyšší kapacity akumulátoru v případě nezapojování článků do série.

V rámci firmware by mohly nastat změny: možnost odesílání dat v periodách odlišných od period měření, optimalizace spotřeby elektrické energie revidováním firmware a zdrojů hodinových signálů, vytvoření pracovního režimu pro možnost konfigurace IQRF sítě skrze USB, případně vytvoření pracovního režimu pro možnost konfigurace GSM modulu skrze USB. V poslední řadě by bylo možno rozšířit komunikátor o možnost využívat také MQTT protokol.

Literatura

- [1] BRTNÍK, Bohumil a David MATOUŠEK. *Algoritmy číslíkového zpracování signálů*. Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-400-2.
- [2] Cypress Semiconductor. *CY15B104Q, 4-Mbit (512 K × 8) Serial (SPI) F-RAM* 2017. [online] Dostupné z: <https://www.cypress.com/file/209146/download>
- [3] DOSTÁLEK, Libor a Alena KABELOVÁ. *Velký průvodce protokoly TCP/IP a systémem DNS*. 2. vyd. Computer Press Praha, 2000. 426 s. ISBN 80-7226-323-4.
- [4] JACKO, Róbert. *Energeticky soběstačný bezdrátový modul pro senzorické aplikace*. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. Dostupné také z: <http://hdl.handle.net/11012/32913>.
- [5] MICRORISC s.r.o. *IQMESH animations* [online] [cit. 3.12.2017]. Dostupné z: <https://iqr.org/technology/iqmesh/iqmesh-animations>
- [6] MICRORISC s.r.o. *IQRF Cloud* [online] [cit. 3.12.2017]. Dostupné z: <https://iqr.org/technology/iqr-cloud>
- [7] MICRORISC s.r.o. *IQRF Cloud Technical guide*. 2015. [online] Dostupné z: <http://www.iqr.org>.
- [8] MICRORISC s.r.o. *IQRF DPA Framework Technical Guide v3.02*. 2017. [online] Dostupné z: <http://www.iqr.org>.
- [9] MICRORISC s.r.o. *IQRF OS Operating System Version 4.02D for TR-7xD - Reference Guide*. 2017. [online] Dostupné z: <http://swww.iqr.org>.
- [10] MICRORISC s.r.o. *IQRF OS Operating System Version 4.02D for (DC)TR-7xD - User's Guide*. 2017. [online] Dostupné z: <http://swww.iqr.org>.
- [11] MICRORISC s.r.o. *IQRF Quick Start Guide For IQRF OS v3.07D and higher*. 2017. [online] Dostupné z: <http://www.iqr.org>.
- [12] MICRORISC s.r.o. *IQRF SPI Technical guide For TR-7xD For IQRF OS v4.02 or higher*. 2017. [online] Dostupné z: <http://www.iqr.org>.
- [13] MICRORISC s.r.o. *Transceiver modules* [online] [cit. 3.12.2017]. Dostupné z: <https://iqr.org/products/transceivers>
- [14] MICRORISC s.r.o. *Two application approaches* [online] [cit. 3.12.2017]. Dostupné z: <https://iqr.org/technology/2-application-approaches>

- [15] MICRORISC s.r.o. *IQRF - Technology for wireless* [online] [cit.26.4.2019]. Dostupné z: <https://www.iqrf.org/products/development-tools/development-kits/ck-usb-04a>
- [16] NXP Semiconductors. *Arm® Cortex®-M0+/Ultra-Low Power Kinetis KL2x USB MCU/NXP* [online] [cit. 25.12.2017]. Dostupné z: <https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/l-seriesultra-low-powerm0-plus/kinetis-kl2x-48-mhz-usb-ultra-low-power-microcontrollers-mcus-based-on-arm-cortex-m0-plus-core:KL2x>
- [17] NXP Semiconductors. *Kinetis KL28Z Data sheet*. 2016. [online] Dostupné z: <https://www.nxp.com/docs/en/data-sheet/MKL28Z512Vxx7.pdf>
- [18] NXP Semiconductors. *Kinetis KL28Z Reference Manual*. 2016. [online] Dostupné z: <https://www.nxp.com/docs/en/reference-manual/MKL28ZRM.pdf>
- [19] NXP Semiconductors. *Kinetis L Series MCUs Selector Guide An Ultra-Low-Power Series Based on Arm® Cortex®-M0+ Cores*. 2015. [online] Dostupné z: https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/l-seriesultra-low-powerm0-plus/kinetis-kl2x-48-mhz-usb-ultra-low-power-microcontrollers-mcus-based-on-arm-cortex-m0-plus-core:KL2x?tab=Documentation_Tab
- [20] Quectel Wireless Solutions. *GSM Quectel Cellular Engine GSM TCPIP Recommended Process*. 2015. [online] Dostupné z: https://www.quectel.com/UploadImage/Downlad/Quectel_GSM_TCPIP_Recommended_Process_V1.2.pdf
- [21] Quectel Wireless Solutions. *Quectel GSM/GPRS M95* [online] [cit. 6.4.2019]. Dostupné z: <https://www.quectel.com/product/m95.htm>
- [22] Quectel Wireless Solutions. *M95 AT Commands manual*. 2014. [online] Dostupné z: <https://www.quectel.com/product/m95.htm>
- [23] Quectel Wireless Solutions. *M95 Hardware Design*. 2013. [online] Dostupné z: https://www.quectel.com/UploadImage/Downlad/M95_Hardware_Design_V1.3.pdf
- [24] Texas Instruments. *bq28z610 Impedance Track™ Gas Gauge and Protection Solution for 1-Series to 2-Series Cell Li-Ion Battery Packs* 2017. [online] Dostupné z: <https://www.ti.com/lit/ds/symlink/bq28z610.pdf>
- [25] Texas Instruments. *bq28z610 Technical Reference Manual (Rev. C)* 2018. [online] Dostupné z: <http://www.ti.com/lit/ug/sluya65c/sluya65c.pdf>
- [26] Texas Instruments. *bq28z610EVM 1- to 2-Series Li-ion Battery Pack Manager Evaluation Module* 2015. [online] Dostupné z: <http://www.ti.com/lit/ug/slube3/slube3.pdf>